Autenticación de Imágenes Basada en la Extracción de Características

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
CURSO 2017–2018

Víctor Rodríguez Carreño

Directores

Luis Javier García Villalba
Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Febrero de 2018

Agradecimientos

En primer lugar quiero dar las gracias a mis directores Luis Javier García Villalba y Ana Lucila Sandoval Orozco por haberme brindado la oportunidad de trabajar en un proyecto de tal magnitud. También por haberme guiado a lo largo del mismo respondiendo a mis dudas y dándome facilidades para realizar un trabajo del que me siento orgulloso.

En segundo lugar quiero agradecer a Esteban Armas Vega por haberme ayudado en todo momento y haber estado siempre ahí para cualquier duda que me pudiese surgir a lo largo de los seis meses.

Por último a quiero dar las gracias a mi familia por su apoyo diario y por sus incansables ánimos en los momentos más complicados del proyecto.

Índice General

Índice de Figuras

Índice de Tablas

Resumen

Con el creciente número de aplicaciones de software que permiten alterar imágenes digitales y su facilidad de uso debilitan la credibilidad de una imagen. Este problema unido a la facilidad de distribución de la información a través de Internet (blogs, redes sociales, etc.), ha provocado que la sociedad tienda a aceptar como cierto todo lo que ve sin cuestionar su veracidad. La falsificación de imágenes se ha convertido en una gran amenaza para la credibilidad de la información y el análisis forense de imágenes tiene como objetivo detectar y localizar falsificaciones de imágenes utilizando múltiples pistas que le permita determinar la veracidad o no de una imagen. Este trabajo propone dos métodos de autenticación de imágenes digitales. El primero mediante el análisis de patrones locales de textura, en este caso, el sistema propuesto combina el patrón binario local con la transformada discreta wavelet y la transformada discreta del coseno para extraer las características de cada uno de los bloques de la imagen investigada. Posteriormente, se utiliza la máquina de soporte vectorial para crear el modelo que permita la verificación de la autenticidad de una imagen. El segundo método propuesto realiza una detección de alteraciones de tipo *copy-move* dentro de una imagen, utilizando para ello la transformada discreta del coseno. Las características obtenidas de estos coeficientes permite obtener vectores de transferencia, los cuales se agrupan y mediante el uso de un umbral de tolerancia permite determinar si existe o no regiones copiadas y pegadas dentro de la imagen analizada. Los resultados obtenidos de los experimentos llevados acabo en este trabajo demuestran lo eficaces de los métodos propuestos. Para la evaluación de los métodos propuestos se realizaron experimentos con bases de datos públicas de imágenes falsificadas que son ampliamente utilizadas en la literatura.

Palabras clave: Análisis Forense, Clasificación, Identificación de Fuente, Imágenes Digitales, Máquina de Soporte Vectorial, PRNU, Respuesta Fotónica No Uniforme, Ruido del Sensor, SVM, Transformada Wavelet.

Abstract

With the increasing number of software applications that allow altering digital images and their ease of use, they weaken the credibility of an image. This problem, together with the ease of distributing information through the Internet (blogs, social networks, etc.), has caused that, normally, the society accept as true everything that it sees without questioning its veracity. Image counterfeiting has become into a major threat to the credibility of information and forensic image analysis is aimed at detecting and locating image forgeries using multiple clues that allows it to determine the veracity or otherwise of an image. This work proposes two methods of digital image authentication. The first one through the analysis of local texture patterns; in this case, the proposed system combines the local binary pattern with the discrete wavelet transform and the discrete cosine transform to extract the characteristics of each of the blocks of the investigated image. Subsequently, the vector support machine is used to create the model that allows verification of the authenticity of an image. The second proposed method performs a detection of copy-move alterations within an image, using the discrete cosine transform. The characteristics obtained from these coefficients allow us to obtain transfer vectors, which are grouped together and through the use of a tolerance threshold, it is possible to determine if there are regions copied and pasted within the analyzed image. The results obtained from the experiments carried out in this work, demonstrate the effectiveness of the proposed methods. For the evaluation of the proposed methods, experiments were carried out with public databases of falsified images that are widely used in the literature.

Keywords: Acquisition Source Identification, Classification, Digital Image, Forensics Analysis, Image Anonymity, Clustering, Image Features, Image Forgery, Mobile Device, Photo Response Non Uniformity, Source, Sensor Noise, Support Vector Machine, Theia, Wavelet Transform.

Capítulo 1

Introducción

1.1. Motivación

En los últimos años el uso de dispositivos móviles ha crecido considerablemente llegando a ser una herramienta que forma parte de la vida cotidiana de la sociedad actual. Debido a este incremento de dispositivos, el tráfico de datos también ha aumentado creando una red de comunicación cada vez más grande entre los usuarios que facilita compartir datos de forma masiva y rápida. Entre los datos compartidos están las imágenes digitales que gracias a las redes sociales y a las aplicaciones de mensajería instantánea, se han convertido en uno de los principales focos del tráfico de datos.

La mejora continua de las cámaras incorporadas en los dispositivos móviles junto a la evolución de las herramientas de edición de imágenes han hecho que cada vez sea más sencillo manipular una imagen con excelentes resultados. Para hacer frente a este tráfico masivo de imágenes manipuladas el área de análisis forense investiga nuevas técnicas de detección de manipulaciones, para evaluar la integridad de una imagen. Las imágenes manipuladas no solo se encuentran en el tráfico generado por la red sino que llevan existiendo desde hace décadas y están presentes en muchos sectores (política, cine, prensa, etc.).

Debido a la cantidad de información que puede contener una imagen digital se ha usado desde hace muchos años como método principal para influir en la sociedad. Es por ello que en la prensa y la política abunda una gran cantidad de imágenes manipuladas. Un ejemplo de este tipo de imágenes con fines políticos se muestra en la Figura 7.1. En ella se observa como se eliminó al Rey Jorge VI que estaba junto al primer Ministro de Canadá, William Lyonde de la fotografía original. La imagen manipulada fue utilizada en un cartel en las elecciones del primer ministro. Al eliminar al rey de la imagen daba una sensación de más cercanía al pueblo y no tanto a la familia real.

(a) *Imagen manipulada* (b) *Imagen original*

Figura 1.1: Imagen manipulada con fines políticos [FT17]

No todas las imágenes manipuladas persiguen fines políticos o ideológicos, este tipo de imágenes es muy común en las redes sociales donde solamente se busca la popularidad y que llegue al mayor número de usuarios posible. Un ejemplo de foto viral que causó gran repercusión tuvo lugar poco después del 11 de septiembre de 2001. La imagen de la Figura 1.2 se difundió bajo el contexto de ser la última fotografía tomada tras el atentado de las torres gemelas. Debido al clima del día del atentado y a varias observaciones sobre la posición física desde la que se hizo la fotografía se demostró que había sido manipulada.

Figura 1.2: Imagen viral manipulada del atentado de las torres gemelas [Lis07]

Además de los sectores antes mencionados las imágenes digitales manipuladas han tenido importancia en otras áreas con objetivos muy diferentes. Por ejemplo, en el sector judicial las imágenes han ido ganando importancia debido a que pueden suponer una evidencia de gran valor para la resolución de un juicio. Para que una imagen pueda ser usada como prueba válida o evidencia de algún acto con fines legales se debe asegurar su integridad y demostrar que no ha sido objeto de manipulación. Para llevar a cabo este tipo de autenticación es necesario hacer uso de técnicas robustas de identificación de manipulaciones que puedan garantizar con gran fiabilidad que la imagen es original.

Según avanza la tecnología cada vez es más complicado detectar este tipo de manipulaciones, herramientas como *Photoshop* o *GIMP* permiten la edición de imágenes con resultados altamente profesionales. Este tipo de software requiere de conocimientos previos sobre la edición de imágenes para conseguir manipulaciones difíciles de detectar. Con el incremento del uso de los dispositivos móviles, el tráfico de datos y las cámaras incorporadas de alta resolución cada vez son más las aplicaciones que traen integradas funcionalidades de edición de imágenes. Estas aplicaciones al contrario de las mencionadas anteriormente no requieren de ningún

conocimiento sobre la edición de imágenes para obtener manipulaciones muy difíciles de detectar.

Por todas estas razones el análisis forense de imágenes digitales para dispositivos móviles está teniendo mucha importancia hoy en día. Se deben estudiar y proponer técnicas de identificación que permitan hacer frente al gran número de imágenes manipuladas que existen hoy en día.

1.2. Contexto

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado RAMSES aprobado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (Convocatoria H2020-FCT-2015, Acción de Innovación, Número de Propuesta: 700326)y en el que participa el Grupo GASS del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM).

Además de la Universidad Complutense de Madrid participan las siguientes entidades:

- Treelogic Telemática y Lógica Racional para la Empresa Europea SL (España)
- Ministério da Justiça (Portugal)
- University of Kent (Reino Unido)
- Centro Ricerche e Studi su Sicurezza e Criminalità (Italia)
- Fachhochschule fur Offentliche Verwaltung und Rechtspflege in Bayern (Alemania)
- Trilateral Research Consulting LLP (Reino Unido)
- Politecnico di Milano (Italia)
- Service Public Federal Interieur (Bélgica)
- Universitaet des Saarlandes (Alemania)
- Dirección General de Policía - Ministerio del Interior (España)

1.3. Objetivos

Los objetivos que se han marcado en el presente trabajo son los siguientes:

- Revisar la literatura actual respecto a los usos de imágenes falsificadas para obtener un conocimiento detallado de las técnicas de manipulación usadas hoy en día y su implementación.
- Estudiar el estado del arte sobre las técnicas de detección de manipulaciones de imágenes.
- Diseñar e implementar algoritmos robustos y eficientes que consigan resultados óptimos en la detección de las manipulaciones de imágenes digitales.
- Realizar pruebas que demuestren la robustez y la validez de los resultados obtenidos por los algoritmos.

1.4. Plan de Trabajo

El trabajo se dividió en 7 fases. A continuación se dará una explicación detallada de cada una de las fases que se han llevado a cabo a lo largo del proyecto.

- **Lectura de artículos**: Esta fase consistió en una investigación sobre: el análisis forense de imágenes, las técnicas de falsificación de imágenes y las técnicas de detección de manipulaciones. También se revisaron los trabajos de fin de grado anteriores con el mismo tema común al nuestro para así obtener varias visiones desde otros puntos de vista.
- **Documentación**: La actividad de documentación se llevó a cabo de forma paralela a lo largo de todo el proyecto. De esta forma se evitaba posibles pérdidas de información relevante para la memoria o para el trabajo de desarrollo.
- **Reuniones de control**: Semanalmente se concertaban 2 reuniones para llevar un seguimiento del proyecto. Se trataban los siguientes puntos a realizar y se mostraban los avances llevados a cabo.

- **Preparación del entorno de trabajo**: Esta actividad consistió en preparar el equipo con el que se iba a trabajar. Se procedió a la descarga de las herramientas adecuadas así como a varias comprobaciones para que no hubiese problemas en las fases de desarrollo. Esta actividad se realizó en mayor medida junto a las reuniones de control.
- **Desarrollo**: En esta fase se procedió a llevar a cabo el desarrollo de los algoritmos de este trabajo. A finales de esta actividad se iniciaron las pruebas de código, de esta manera se iban mejorando los algoritmos hasta conseguir los resultados más óptimos.
- **Pruebas**: Esta fase ha sido la última en llevarse a cabo. Se han realizado numerosas pruebas de calidad de los resultados y de eficiencia de los algoritmos. Como puede observarse en el diagrama de Gantt (1.3) esta actividad ha llevado más tiempo que el proceso de desarrollo. Esto se debe a que ha sido de gran importancia para contrastar resultados con investigaciones anteriores.

Para tener una visión sobre el plan de trabajo estructurado por fechas se puede ver el diagrama de Gantt mostrado en la Figura 1.3.

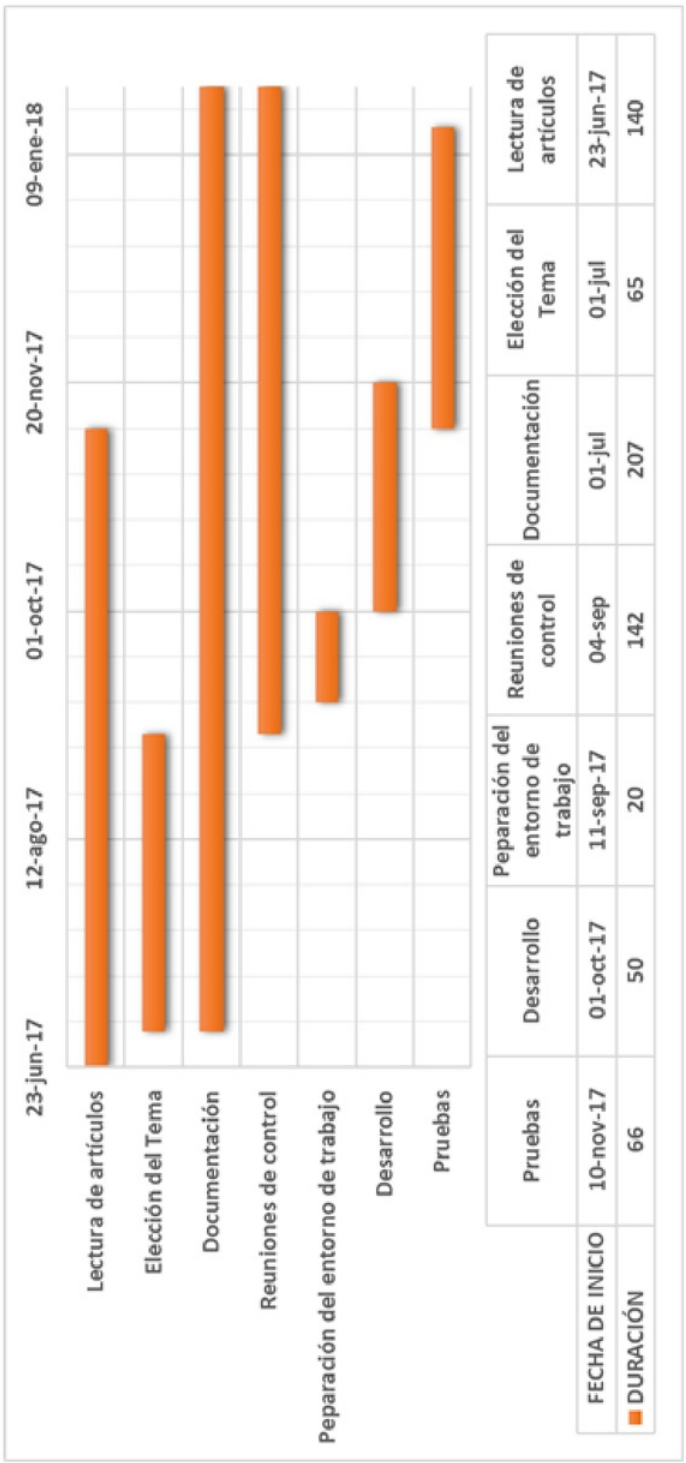


Figura 1.3: Diagrama de Gantt sobre el plan de trabajo

1.5. Estructura del Trabajo

El trabajo consta de 6 capítulos en total, siendo el actual el capítulo de introducción en el que se describe la motivación, los objetivos, el plan y la estructura del mismo. A continuación se hará una breve explicación de lo que trata cada uno de los 5 restantes.

El capítulo 2 hace un recorrido desde los orígenes de la fotografía hasta la actualidad en el que se mostrarán ejemplos de la evolución de las técnicas de manipulación. Para finalizar se detallan los tipos de técnicas y herramientas utilizadas en la manipulación de las imágenes digitales.

El capitulo 3 describe las principales técnicas de detección de imágenes digitales manipuladas, haciendo énfasis en las técnicas con enfoque pasivo más relevantes existentes en la literatura. Se hará referencia a investigaciones relacionadas con cada una de las técnicas de manipulación existentes.

El capítulo 4 trata sobre las contribuciones del presente trabajo. Este capítulo empieza con unas consideraciones generales que ayudarán a comprender algunos detalles concretos de la parte de desarrollo. Se explicarán los algoritmos propuestos en este trabajo.

El capítulo 5 presenta los experimentos que evalúan los algoritmos propuestos en el capítulo 4. Comienza detallando la configuración con la que se han realizado las pruebas y, posteriormente, describe cada uno de los experimentos realizados. Se muestran los resultados y se analizan las mejoras obtenidas y los defectos encontrados.

El capítulo 6 recoge las conclusiones finales del trabajo. En el también se proponen mejoras para trabajos futuros.

Finalmente, en los capítulos 7 y 8 se encuentran la introducción y conclusiones en inglés

Capítulo 2

Manipulación de Imágenes Digitales

El objetivo de este capítulo es explicar los diferentes conceptos generales relacionados con la falsificación de imágenes digitales, así como los tipos de técnicas y herramientas utilizadas en la manipulación de las mismas. A si mismo se hará un recorrido desde los orígenes de la fotografía hasta la actualidad mostrando ejemplos de la evolución de las técnicas de manipulación.

Según el informe publicado en febrero de 2017 por Cisco Systems (Cisco) [CIS17] el tráfico de datos móviles se ha multiplicado por 18 en los últimos 5 años y prevé que este tráfico siga en aumento. También se indica en el informe que para el año 2021 las velocidades de conexión de la red móvil se triplicarán y habrá 1.5 dispositivos móviles por habitante, a su vez se prevé que para este año más de las tres cuartas partes del tráfico mundial de datos móviles serán de vídeo. En la Figura 2.1 se representa una predicción del aumento del tráfico generado por dispositivos móviles entre 2016 y 2021, en ella se observa que se espera un aumento del tráfico total de datos móviles a 49 exabytes por mes.

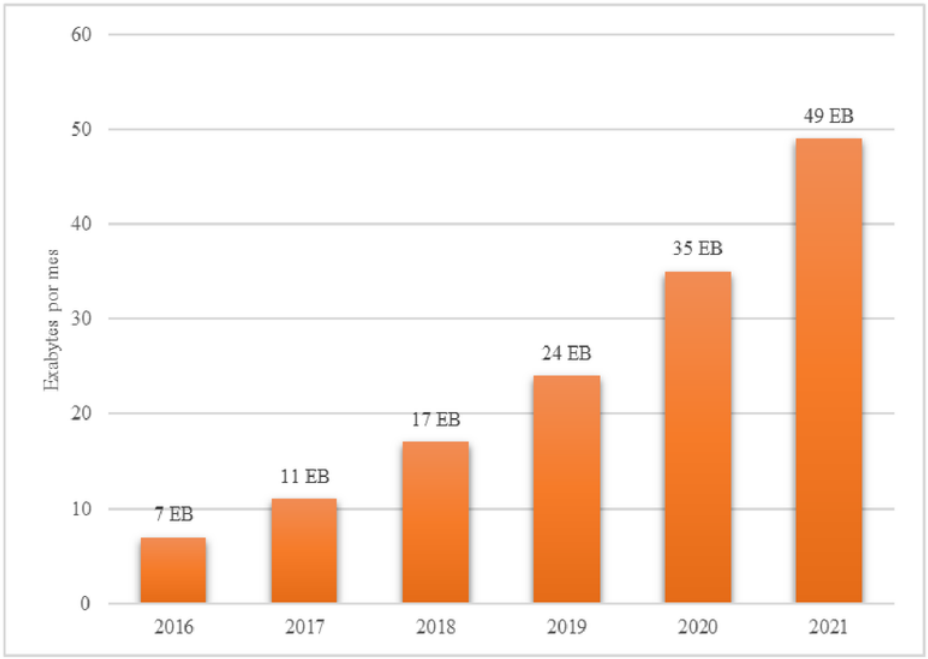


Figura 2.1: Predicción del aumento del tráfico en dispositivos móviles

En la Figura 2.2 se puede observar el aumento entre 2016 y 2021 de los dispositivos móviles inteligentes. Para hacer frente a los recursos informáticos y capacidades de conexión de estos dispositivos inteligentes. Como se observa en la Figura 2.2 a su vez hay un aumento de la demanda de redes más capacitadas e inteligentes. Los dispositivos y conexiones inteligentes tienen capacidades avanzadas de computación y multimedia con un mínimo de conectividad 3G. La proporción de dispositivos inteligentes y conexiones como porcentaje del total aumentará del 46 % en 2016, al 75 %, para el año 2021.

Como consecuencia, los dispositivos móviles inteligentes ocuparán la mayor parte del tráfico y demanda. A su vez, las mejoras de calidad en las conexiones que se prevén van a permitir que se pueda compartir grandes cantidades de contenido multimedia de forma rápida y sencilla. Sin embargo esta facilidad de crear contenido multimedia y compartirlo puede convertirse en un problema si lo sumamos a la facilidad con la que hoy en día es posible encontrar numerosas herramientas dedicadas a la manipulación de imágenes digitales; con el paso de los años este tipo de software ha ido evolucionando ofreciendo mejoras en los resultados finales y reduciendo su dificultad, uso y configuración. Esto ha permitido que usuarios

sin conocimientos mínimos sobre edición de imágenes digitales puedan realizar manipulaciones rápidamente y con resultados profesionales.

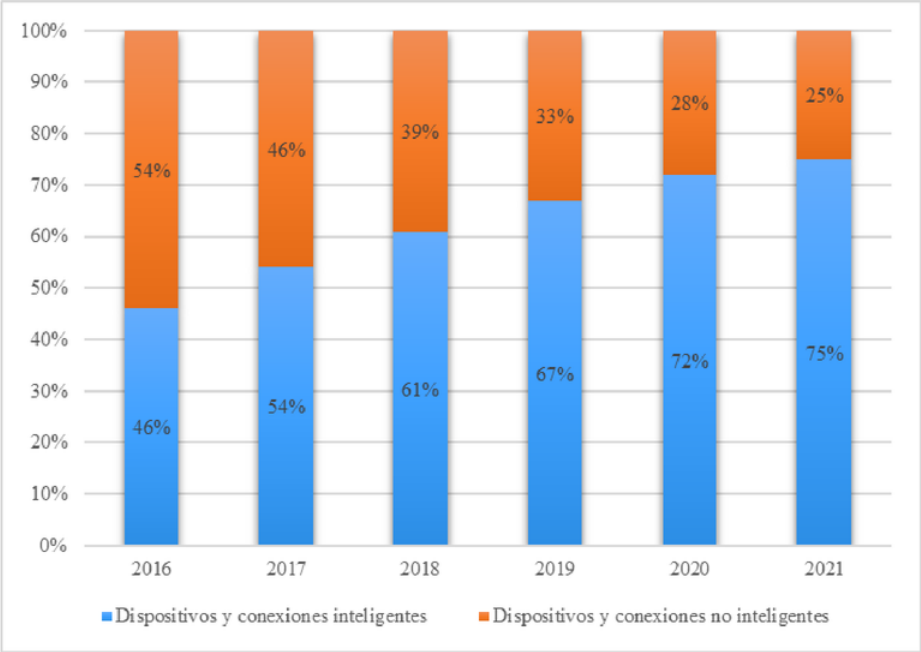
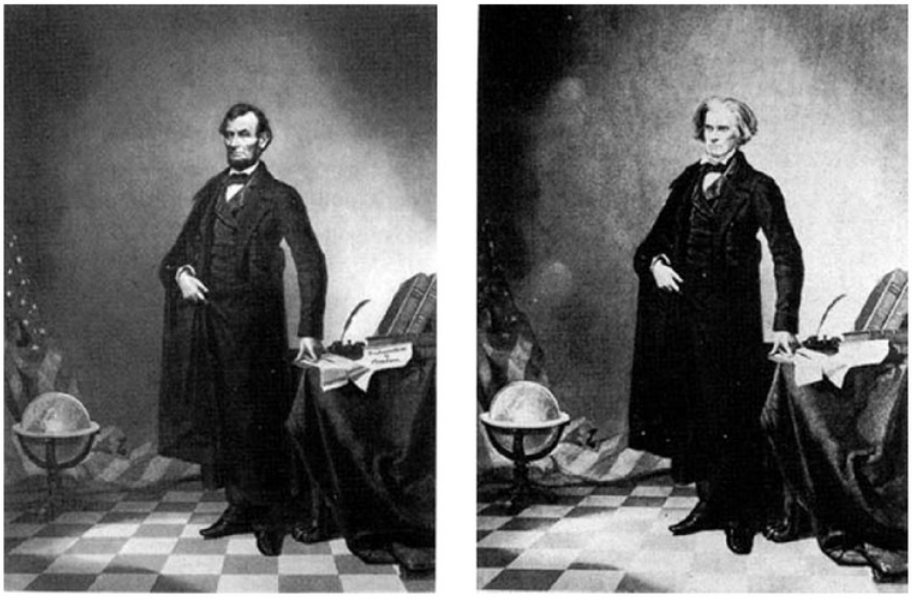


Figura 2.2: Predicción del aumento de dispositivos móviles y conexiones inteligentes

La evolución del software de edición de imágenes supone un gran problema en el ámbito legal, se debe garantizar la integridad de las imágenes o vídeos digitales para que sean válidas como evidencias. En ocasiones una simple imagen o vídeo puede determinar la resolución de un juicio debido a que aportan una gran cantidad de información. Un ejemplo de caso donde se usó un vídeo digital como prueba relevante fue publicado por el periódico *el Mundo* [Mun17]. El caso acabó con la detección de un conductor que circulaba a más de 200 kilómetros por hora. Un ciudadano encontró un vídeo donde se mostraba al conductor circulando a esa velocidad e inmediatamente lo denunció a la policía. En este caso, el vídeo es un claro ejemplo de evidencia digital y se debe contrastar su autenticidad e integridad para demostrar que fue así como ocurrieron los hechos.

2.1. Historia de la Manipulación de Imágenes

La falsificación de imágenes empezó a llevarse a cabo pocos años después de que Joseph Nicéphore crease la primera fotografía en el año 1826. Un primer ejemplo de manipulación fue registrado a principios de la década de 1860, cuando una foto de Abraham Lincoln fue compuesta insertando la cabeza de Lincoln en el cuerpo de un retrato de John C. Calhoun (Figura 2.3).



(a) *Imagen manipulada* (b) *Imagen original*

Figura 2.3: Foto manipulada de Abraham Lincoln [FT17]

En 1920 Joseph Stalin hizo uso del retoque fotográfico con fines propagandísticos. Como se puede observar en la Figura 2.4 se hizo desaparecer al comisario para asuntos internos Nikolai Yezhov tras ser ejecutado en 1940.

Otra foto histórica manipulada tuvo lugar en 1937, Adolf Hitler hizo quitar a Joseph Goebbels de la fotografía original (Figura 2.5). Todavía no está claro por qué se eliminó a Goebbels de la imagen.

(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.4: Manipulación fotográfica de Stalin [FT17]

(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.5: Desaparición de Joseph Goebbels de la imagen original [FT17]

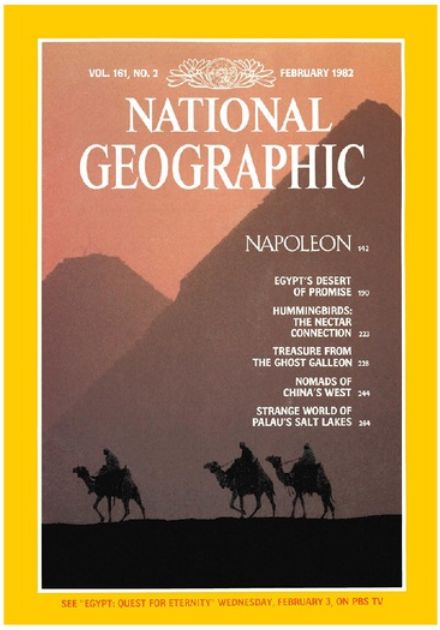
Benito Mussolini mandó quitar en 1942 a un ayudante que se encontraba en la fotografía original, de esta forma lograba un retrato más heroico de sí mismo y a su vez un mejor mensaje propagandístico (Figura 2.6).

En la década de 1980 empezó a surgir la era digital y a su vez aparecieron las primeras herramientas de edición de imágenes digitales. Entre las primeras manipulaciones que se conocen de este periodo destaca la realizada por National Geographic en la portada de su revista sobre Egipto, tuvo lugar en 1982 (ver Figura 2.7). Gordon Gahen tomó una imagen horizontal de las Grandes Pirámides de Giza, estas tuvieron que ser escaladas para ajustarse al formato vertical de la revista. El fotógrafo que realizó la imagen se quejó a la revista sobre la manipulación de su imagen.

(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.6: Imagen propagandística de Benito Mussolini [FT17]



(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.7: Portada de la revista National Geographic manipulada [FT17]

Otro ejemplo famoso de manipulación relacionada con revistas data del 1989. La portada de TV Guide muestra esta imagen de la presentadora de programas de entrevistas diurna Oprah Winfrey. Esta imagen fue creada empalmando la cabeza de Winfrey en el cuerpo de la actriz Ann-Margret (Figura 2.8(a)), tomada de una toma de publicidad de 1979 (Figura 2.8(b)). El compuesto fue creado sin el permiso

de Winfrey y Ann-Margret.

(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.8: Portada de la revista TV Guide [FT17]

En la década de 1990 llegaron las cámaras digitales de alta resolución, las potentes computadoras personales y los programas sofisticados de edición de fotos, cada vez se manipulaban fotografías con mayor frecuencia. A su vez, detectar las manipulaciones se convertía en una tarea más costosa. Como ejemplo de esta década se destaca la manipulación realizada por la revista Time, en 1994. En la portada de la revista puede observarse una imagen de Simpson poco después de ser arrestado por asesinato. Esta imagen fue manipulada digitalmente desde la fotografía original que apareció, inalterada, en la portada de Newsweek (Figura 2.9(b)). Tiempo después se acusó a la revista de manipular la fotografía para hacer que Simpson pareciese más amenazante y peligroso, como se observa en la Figura 2.9(a)

(a) *Imagen manipulada* (b) *Imagen original*

Figura 2.9: Imagen de Simpson en la portada de la revista Time [FT17]

En la actualidad es complicado destacar imágenes manipuladas debido al gran volumen de éstas que circulan por la red sin ningún tipo de filtro. Desde la llegada de las redes sociales se ha hecho posible que cualquier persona pueda compartir rápidamente una fotografía con un gran número de usuarios. Es muy habitual encontrar un predominio de imágenes manipuladas en estas redes ya que permiten a los usuarios causar un mayor impacto sobre sus seguidores. Además algunas de estas redes proporcionan herramientas integradas de manipulación de imágenes que permite la edición justo después de capturar la fotografía.

2.2. Técnicas de Manipulaciones de Imágenes Digitales

A continuación se presentan los tipos de manipulaciones existentes:

2.2.1. Retoque de Imágenes

El retoque de imágenes consiste en aplicar diferentes filtros sobre la imagen original para mejorarla según unos objetivos manteniendo siempre unas características similares. Para ello se copian y pegan regiones de la imagen de la misma área. Los retoques que se realizan suelen estar enfocados a perfeccionar el acabado de las imágenes, el acabado varía dependiendo del contenido de la imagen y de los fines con los que se realiza la manipulación.

Esta técnica de manipulación es muy común en los sectores de la publicidad, cine y comunicación. Por ejemplo es habitual que en revistas de moda las imágenes hayan pasado por alguna técnica de retoque para camuflar desperfectos y así aumentar los niveles de belleza en la fotografía. En la Figura 2.10 podemos observar los retoques realizados en la portada de una revista llamada *Nitro*. En ella puede observarse la apariencia física real de la actriz (Figura 2.10(b)) y el resultado final tras realizar la técnica de manipulación (Figura 2.10(a)).

(a) *Imagen manipulada* (b) *Imagen original*

Figura 2.10: Portada manipulada de la revista Nitro [Sun14]

2.2.2. Copia-Pega

La técnica de copia-pega consiste en copiar una región y pegarla encima de otra región de la misma imagen. El área copiada puede ser pegada en cualquier parte de la imagen. Esta es la principal diferencia con la técnica de retoque en la cual la región pegada proviene de una región próxima. Esta técnica permite ocultar partes de la imagen o duplicar elementos y regiones. También pueden llevarse a cabo técnicas de post-procesamiento, como escalar, rotar o aplicar alguna clase de filtro, estas técnicas hacen más costoso el proceso de detección de la manipulación. Se usa habitualmente para ocultar información relevante de una o más áreas de la imagen.

Como ejemplo de este tipo de técnica de manipulación se presenta en la Figura 2.12 una foto histórica que tuvo lugar en Irán en el año 2008. En la Figura 2.11(a) se puede observar la foto publicada por la agencia de noticias iraní (*Sepah News*), en ella se mostraba el lanzamiento de cuatro misiles con éxito. La foto original que se publicó posteriormente se muestra en la Figura 2.11(b) donde realmente se ve que fueron tres los misiles que habían sido lanzados con éxito. También se aprecia que se han usado técnicas de postprocesado en el humo expulsado por el misil para camuflar la manipulación.

(a) *Imagen manipulada*

(b) *Imagen original*

Figura 2.11: Foto manipulada del lanzamiento de misiles iraní [FT17]

2.2.3. Empalme

La técnica de empalme consiste en copiar una región de una determinada imagen y pegarla en otra distinta, de tal forma que se mezclan ambas imágenes creando una sola. Es muy usada en fotomontajes donde se combinan dos imágenes dando la sensación de ser una sola. Detectar el área exacta que se ha falsificado en la imagen mediante la técnica de empalme es de gran complejidad en comparación a las anteriores técnicas de manipulación. Esto se debe a que no es posible buscar áreas duplicadas ya que la región manipulada proviene de una imagen diferente.

En la Figura 4.1 se muestra un ejemplo en el que se pueden ver dos imágenes originales que se han usado para crear una foto manipulada por empalme. De la primera imagen mostrada en la Figura 2.12(b) se ha recortado el letrero del templo y se ha pegado en la segunda imagen de la Figura 2.12(b) para así generar el resultado final que puede verse en la Figura 2.12(a).

(a) *Imagen manipulada* (b) *Imágenes originales*

Figura 2.12: Foto manipulada del lanzamiento de misiles iraní [FT17]

2.2.4. Modificación o Eliminación de la Huella Digital

Durante el proceso de generación de una imagen es posible que se introduzcan algunos defectos que se vean reflejados como ruido en la imagen final. Los defectos son producidos por los sensores del dispositivos y son de gran ayuda para identificar la cámara que generó una imagen determinada.

Existen diversas fuentes de imperfecciones y ruido introducidas en las diferentes etapas del proceso de generación de la imagen en la cámara. Incluso si se toma una fotografía uniforme y completamente iluminada es posible observar pequeños cambios de intensidad entre los píxeles. Esto se debe al ruido de disparo que es aleatorio y, en gran parte, al patrón de ruido que es determinista y se mantiene aproximadamente igual si se toman varias fotografías de la misma escena.

Hay dos tipos de ruido que son importantes. El primer tipo de ruido es causado por defectos en el sensor de la cámara. Estos incluyen defectos puntuales, defectos de punto caliente, píxeles muertos, efectos inesperados, entre otros. Estos defectos hacen que los valores de los píxeles en la imagen se desvíen en gran medida. Por ejemplo, los píxeles muertos aparecen como puntos negros en la imagen y los defectos de puntos calientes aparecen como píxeles muy brillantes. El patrón de ruido en una imagen se refiere a cualquier patrón espacial que no cambia de una imagen a otra. Las corrientes oscuras son corrientes perdidas desde el sustrato del sensor hacia los píxeles individuales. Esto varía de píxel a píxel y la variación se conoce como ruido de patrón fijo (FPN). El ruido PRNU está formado principalmente por la uniformidad de píxel PNU y los defectos de baja frecuencia como la configuración del zoom y la refracción de la luz en las partículas de polvo y lentes. Este patrón es muy importante para detectar la fuente de procedencia de una imagen ya que cada dispositivo tendrá asociado un patrón de ruido diferente [KMC+07][GVSORCHC17]. En la Figura 2.13 podemos ver un esquema general de los tipos de patrón de ruido del sensor.

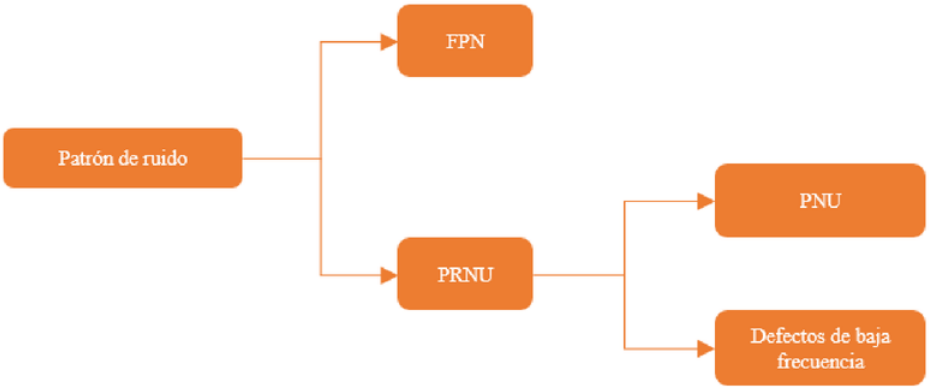


Figura 2.13: Tipos de patrón de ruido del sensor

Las técnicas de manipulación que hacen uso de la huella digital se diferencian de las anteriores en que no se centran en la parte gráfica de la imagen si no en

la información que esta contiene. Estas técnicas pueden dividirse en dos tipos que persiguen objetivos diferentes [GVSORCHC17], estos tipos son:

- **Anonimización de la imagen**: Consiste en eliminar la información de la fuente de origen de la cual procede la imagen. Esta información también conocida como fuente de la imagen es la relación entre un dispositivo y la imagen que genera. De esta forma se puede llegar a identificar el individuo propietario del dispositivo que genero dicha imagen, por ello es de gran importancia conocer esta información. Para realizar este tipo de técnica se elimina el PRNU de la imagen que se quiere modificar, así se consigue que la imagen resultante sea de procedencia anónima.
- **Falsificación de la imagen**: Se basa en realizar una modificación sobre el origen de la imagen, de esta manera se consigue cambiar la información real sobre el modelo y la marca del dispositivo que generó la imagen digital. Un uso de esta técnica muy común consiste en extraer la huella digital de una imagen (PRNU) y sustituir la huella de otra imagen por la extraída. De esta manera se consigue falsificar la información sobre el origen cambiando el dispositivo que realmente generó la imagen por la huella de otra que no lo hizo. La mayoría de técnicas anti-forenses de identificación de la fuente de origen hacen uso de este patrón.

2.3. Herramientas de Manipulación de Imágenes

Con el paso de los años las herramientas de edición de imágenes se han ido perfeccionado, ofreciendo mejores resultados y simplificando su funcionalidad.

Las herramientas de edición de imágenes más destacadas a lo largo de la historia son las siguientes:

- **Microsoft Paint**: Esta herramienta fue desarrollada en el año 1982 por la empresa *Microsoft*. Viene instalada por defecto en los sistemas operativos Microsoft Windows y es una herramienta sencilla con funciones limitadas. Soporta los formatos PNG, BMP, GIF, TIFF y JPEG.
- **Adobe Illustrator**: Herramienta desarrollada en el año 1986 que permite crear y editar imágenes en multitud de formatos como SVG, PNG, BMP, GIF,

TIFF, JPEG, entre otros. También permite añadir diferentes efectos como la modificación de color y textura o efectos en tres dimensiones. Esta herramienta se adquiere mediante la licencia *Adobe CLUF*.

- **Photoshop**: Es uno de los programas más conocidos y líder en el mercado del área de la edición de imágenes. Está desarrollado por la empresa *Adobe Systems* y se creó en el año 1991. Esta herramienta se usa principalmente para el retoque de fotografías y gráficos. Permite dividir una imagen en varias capas y trabajar con ellas de forma independiente, facilitando la aplicación de efectos, filtros, textos y múltiples tratamientos a diferentes áreas sin afectar al resto de la imagen. Photoshop ha ido extendiéndose por diversos sectores relacionados con el mundo de la fotografía y es sin duda un estándar para el retoque fotográfico. Se adquiere mediante la licencia *Adobe CLUF* y soporta multitud de formatos como BMP, JPEG, PNG, GIF, PSD, entre otros.
- **GIMP**: La primera versión de este programa de manipulación de imágenes salió en el año 1996. Su principal característica es el uso de plugins desarrollados libremente por la comunidad. Esto permite que la aplicación siga ampliando su funcionalidad añadiendo nuevos filtros y efectos a las imágenes o creando nuevas herramientas dentro de GIMP. Su licencia es gratuita y soporta múltiples formatos como JPEG, GIF, PNG, TIFF, PSD, SVG, entre otros.

Todas las aplicaciones mencionadas requieren de un proceso de instalación. A partir del año 2000 empezaron a surgir herramientas online alojadas en sitios web o en la nube, permitiendo su uso sin previa instalación en cualquier plataforma. Una de las más destacadas es *Pixrl*. Esta aplicación de manipulación de imágenes surgió en el año 2008. Su principal característica es que no requiere una instalación previa, permite la edición de imágenes de forma online. Ofrece multitud de filtros, efectos y herramientas, así como una interfaz muy intuitiva. Es una de las aplicaciones web más potentes que podemos encontrar hoy en día.

Debido al incremento de dispositivos móviles inteligentes con cámaras de alta calidad empezaron a surgir herramientas de edición de imágenes para este tipo de dispositivos. Una de las características de estas herramientas es la facilidad de uso y las pocas funciones que traen por defecto. Esto se debe a que la mayoría de los usuarios de dispositivos móviles no tienen conocimiento sobre la edición de imágenes. Otra característica es la posibilidad de editar una foto nada más hacerla con la cámara del dispositivo, esto aumenta la rapidez del proceso de edición. Este tipo de aplicaciones son las más usadas hoy en día respecto a las de escritorio o a las

basadas en sitios web y nube, esto es causado por el incremento de dispositivos inteligentes. Muchas de estas herramientas como *Snapseed* o *Photo Lab* pueden descargarse de una plataforma de aplicaciones o bien pueden venir ya incluidas en las redes sociales. Un ejemplo de estas redes son *Instagram* o *Twitter* que traen por defecto la posibilidad de añadir filtros o realizar retoques rápidamente antes de publicar una imagen.

Capítulo 3

Técnicas Forenses de Detección de Manipulación de Imágenes

Como una imagen digital contiene una gran cantidad de información, es posible observar y capturar los diferentes cambios a la hora de alterar los datos de la misma. Sobre esta base surgen las técnicas de detección de manipulaciones. En este capítulo se describen las principales técnicas de análisis forense de imágenes digitales analizando con mayor profundidad las técnicas de detección de manipulaciones de empalme y copia-pega. También se explicarán diferentes conceptos generales que ayudarán a entender el proceso de las diferentes técnicas de manipulación.

Existen dos enfoques forenses diferentes para la detección de manipulaciones de imágenes digitales:

- **Enfoque activo**: Consiste en analizar las marcas de agua o señales que deja un dispositivo de captura en el momento de grabar una imagen digital. La autenticidad se comprueba detectando cambios en estas marcas de agua. El mayor inconveniente de este tipo de enfoque es que muchas cámaras no tienen la capacidad de incorporar este tipo de marcas o firmas, por lo que su alcance es limitado.
- **Enfoque pasivo**: Se basa principalmente en analizar el contenido y las características de la imagen digital. Tiene un alcance muy amplio ya que no necesita de ninguna marca de agua o información previa sobre las imágenes. Para realizar el análisis forense de las imágenes digitales estas técnicas son de

vital importancia. A su vez este enfoque puede clasificarse en: métodos basados en aprendizaje y métodos basados en bloques.

- **Métodos basados en aprendizaje**: Esta categoría es capaz de detectar cualquier tipo de falsificación, se basan en determinar si una imagen ha sido manipulada o no basándose en un entrenamiento realizado previamente. Para las aplicaciones como las redes sociales donde es suficiente verificar si una imagen es original o manipulada se utilizan métodos basados en el aprendizaje debido a su gran capacidad de clasificar grandes conjuntos de datos con un coste de tiempo y recursos reducido.
- **Métodos basados en bloques**: Esta última categoría de métodos detecta la falsificación localizando las regiones que han sido adulteradas mediante copiar y pegar. Los métodos basados en bloques son útiles para presentar pruebas en salas de tribunales o reclamaciones de seguros, pero su mayor inconveniente es que consumen mucho tiempo y recursos, esto los hace no ser adecuado para aplicaciones como las redes sociales, donde se comparte una gran cantidad de imágenes todos los días.

En los últimos años se han propuesto muchos métodos pasivos para la detección de falsificaciones. A continuación se presentan las propuestas de enfoque pasivo más relevantes.

3.1. Detección de Retoque

Como se mencionaba en el capítulo 2 el retoque de imágenes suele estar aplicado a los sectores de la publicidad, cine y comunicación. Es por ello que muchas de las técnicas existentes hoy en día para detectar el retoque de imágenes estén basadas en identificar manipulaciones realizadas en el cuerpo o en la cara del sujeto que aparece en la imagen.

En [KF11] se propuso un algoritmo eficiente diseñado específicamente para predecir la presencia de retoques en imágenes tales como portadas de revistas. En este trabajo se recolectó un conjunto de 468 fotos originales y retocadas de múltiples fuentes en linea. A continuación se pidió a varias personas que clasificaran la cantidad de alteración fotográfica en una escala de 1 (muy similar) a 5 (muy diferente). Con una foto original y retocada se calcularon las modificaciones geométricas y fotométricas. De estas modificaciones se extrajeron ocho estadísticas de resumen que

incorporan el grado de retoque fotográfico. Finalmente, las calificaciones realizadas por las personas se correlacionaron con las estadísticas de resumen utilizando una SVM y de esta manera se pudo obtener qué característica era crítica para determinar el grado con el que una imagen había sido retocada. La precisión máxima obtenida con los experimentos fue de 98,75 %.

La detección de retoques de imágenes también se aplica al sector de las redes sociales, en estas aplicaciones la mayoría de imágenes de perfil muestran únicamente la cara. Algunos usuarios pueden hacer uso de la técnica de retoque para impedir su identificación mostrando una cara falsificada. Es por ello que se necesita de algoritmos robustos que permitan obtener información a nivel facial para determinar una posible manipulación.

Un ejemplo de algoritmo es el propuesto en [BSVB16], que presenta la primera base de datos de caras retocadas. La base de datos "*ND-IIITD retouched faces*", almacena imágenes de 325 sujetos (en su mayoría caucásicos) e imágenes retocadas con una herramienta de edición. Se propuso una red neuronal para extraer características y una *Máquina de Vector de Soporte* (SVM) para clasificar las imágenes en una clase sin retoques o retocada. En los experimentos se obtuvo una precisión de más del 87 % en la clasificación de imágenes como originales o retocadas.

Un problema importante relacionado con las imágenes faciales es el maquillaje virtual. Este tipo específico de retoque genera una simulación de maquillaje real en una cara pudiendo llegar a dificultar el proceso de identificación facial de una persona. Para la detección automática de maquillaje se han propuesto varios métodos.

En [CDR13] se propone la extracción de las características de color, forma y textura de tres regiones faciales predefinidas que se clasifican utilizando SVM con núcleo RBF. Realizan experimentos utilizando la base de datos de YMU para el entrenamiento y la base de datos MIW para realizar las pruebas. La precisión que consiguieron con este método fue de hasta un 93 %.

Posteriormente, en [KAD15] se propuso un algoritmo más preciso para la detección de maquillaje en los mismos conjuntos de datos utilizando características de textura y forma. La técnica propuesta extrae un vector de características que captura las características de forma y textura de la cara usada como entrada del algoritmo. Consiguieron aumentar la precisión a un 98.5 % usando un clasificador SVM.

3.2. Detección de Empalme

El empalme de imágenes es un truco muy común y simple en la manipulación de imágenes. En la era digital actual es muy fácil manipular imágenes digitales sin dejar rastros visibles evidentes, las tareas de post-procesado dificultan en mayor medida la detección de este tipo de falsificaciones. En resumen, la detección de empalme puede considerarse como un problema de detectar una señal débil (artefacto de empalme) en el fondo de una señal fuerte (contenido de imagen). Los distintos algoritmos que usan las técnicas de detección de empalme pueden dividirse en diferentes procesos comunes a todos ellos, estos procesos se muestran en la Figura 3.1.

Como se observa en el esquema propuesto este tipo de técnicas hacen uso del método de aprendizaje, entrenando un gran conjunto de imágenes originales y manipulada, para finalmente realizar una rápida clasificación sobre las imágenes propuestas para analizar. El principal factor diferenciador entre estas técnicas es la forma en que se modelan los cambios estructurales introducidos por la manipulación.

Algunos investigadores usaron DCT para modelar los cambios de manipulación y otros hicieron uso de DWT u otro tipo de procesos.

Cuando se realiza una manipulación de empalme, se altera la distribución local de los patrones de micro bordes al introducir nuevos micro-patrones en el interior de la región pegada. Por tanto, cambia su regularidad y la distribución de frecuencias locales. Todos los métodos que se van a discutir a continuación difieren solo en la forma en que modelan los cambios estructurales causados por la falsificación. El éxito de un método depende de la representación de estos cambios, los cuales serán las características con las que se entrenarán y clasificarán las distintas imágenes.

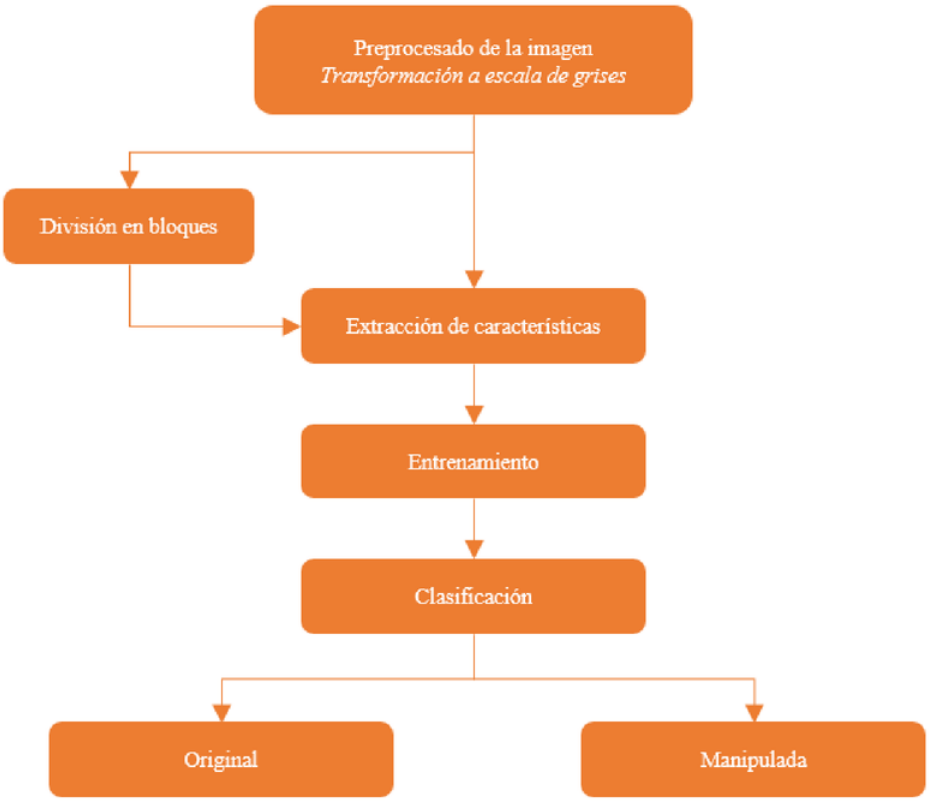


Figura 3.1: Diagrama de procesos de las técnicas de detección de empalme

El método de detección de empalme propuesto en [SCC07] modela los cambios de manipulación utilizando características estadísticas extraídas de matrices 2D generadas al aplicar la transformada discreta de coseno de bloques de varios tamaños (MBDCT). En los experimentos se obtuvo una precisión de 91.40 % y hace un entrenamiento del conjunto de datos "*Columbia*" mediante SVM.

En [ZKR08] se representan los cambios de alteración utilizando características extraídas de matrices 2D generadas al aplicar MBDCT y métricas de calidad de imagen (IQM) y utiliza SVM para la clasificación. Se usa el dataset "*Columbia*" y obtiene una precisión del 87.10 % en los experimentos.

En [ZZ12] usan LBP para extraer características de matrices 2D generadas por MBDCT, PCA para reducción de dimensionalidad y SVM para clasificación. Este método da como resultado una precisión de 89.93 % usando el dataset "*Columbia*".

En [WDT10] se modelan los cambios de la manipulación utilizando la distribución estacionaria del borde de imagen extraída del componente cromático utilizando una cadena de Markov de estado finito y usan SVM como clasificador. Este método logra una precisión máxima de 95.6 % con el dataset "*CASIA TIDE v2.0*".

En [ZL11] exploraron el efecto de diferentes modelos de color en la detección de falsificación de empalme. En este trabajo, se hace una comparación de los modelos cromáticos frente a los modelos RGB y de luminancia utilizados comúnmente. Se emplean cuatro vectores RLRN con diferentes direcciones extraídas de canales de crominancia correlacionados como características para la detección de empalme en imágenes. Finalmente se usa una máquina de soporte vectorial (SVM) como clasificador. Los datasets utilizados en los experimentos son "*CASIA TIDE v1.0*" y "*Columbia*". Esta técnica consigue una precisión de 94.7 %.

[XYS+16] presenta un trabajo para la detección de falsificaciones aplicada a imágenes de huellas digitales, el método que usa emplea DWT junto a LBP, consigue obtener una precisión del 92 %. Se utilizaron para los experimentos diferentes imágenes de huellas dactilares obtenidas por distintos tipos de escáneres. Todas estas imágenes se obtuvieron del dataset "*LivDet*" y se utilizo SVM para su clasificación.

Finalmente [AH17] propone un método que combina el descriptor de textura LBP junto a DCT para detectar cambios producidos por las manipulaciones de empalme y también de copia-pega. Para clasificar realiza un entrenamiento con SVM. Consigue proporcionar la mejor precisión de los métodos anteriormente mencionados, con una tasa de acierto entre el 97.50 % y el 97.77 % en el dataset "*CASIA TIDE v2.0*".

3.3. Técnicas de Identificación de Copia-pega

La técnica de falsificación copia-pega es otro de los métodos más utilizado en la actualidad, en el cual se usan regiones de la imagen para ocultar partes de la misma. La existencia de dos regiones iguales no es común en las imágenes naturales, esta propiedad es explotada para detectar este tipo de manipulaciones. Incluso después de aplicar algunos procesos de post-procesado, como suavizado de bordes, borrosidad y adición de ruido para eliminar las trazas visibles de manipulación, existen dos

regiones extremadamente similares en la imagen manipulada.

Uno de los métodos más utilizados para detectar este tipo de falsificación es utilizar el algoritmo de coincidencia de bloques. En este algoritmo, la imagen se divide en bloques superpuestos y los bloques se comparan para encontrar la región duplicada. En la Figura 3.2 puede verse un esquema general de dicho algoritmo.

Figura 3.2: Diagrama de procesos de las técnicas de detección de copia-pega

En [FSL03] se propuso el primer método basado en la transformada discreta del coseno (DCT) para identificar la falsificación copia-pega en 2003. En este trabajo, la imagen se divide en bloques superpuestos de 16×16, y se utilizan coeficientes DCT para la extracción de características de estos bloques. A continuación, los coeficientes DCT de los bloques se clasifican lexicográficamente. Después de la clasificación lexicográfica, se distinguen cuadrados comparables y se encuentran las regiones duplicadas. En este trabajo, se realizan operaciones de retoque robustas en la imagen, pero no se lleva a cabo ninguna otra prueba de vigor.

[CPF04] propuso un método para identificar regiones duplicadas de imágenes en 2004. Este método aplica PCA en lugar de coeficientes DCT, calcula los valores *Eigen* y los vectores propios de cada bloque de la imagen. Las regiones duplicadas se detectan automáticamente mediante la clasificación lexicográfica. Este algoritmo es una técnica eficiente y robusta para la detección de falsificación de imágenes, incluso si la imagen está comprimida o es ruidosa.

En [KW08] se propuso el uso de SVD para distinguir las áreas alteradas en una imagen digital en 2008. Con SVD se extrae también el vector de características y reduce las dimensiones del mismo. Se identificaron bloques similares mediante el uso de clasificación lexicográfica. Este método resultó ser robusto y eficiente. Los resultados experimentales demuestran la validez del enfoque propuesto para las imágenes manipuladas sometidas a filtros de desenfoque gaussiano, contaminación de ruido y compresiones.

En [HGZ08] se propuso la detección de falsificación de copia-pega en imágenes digitales usando el algoritmo SIFT en el año 2009. En este documento, los autores presentaron el algoritmo de cálculo SIFT utilizando la función de coincidencia de bloques. Este algoritmo ofrece excelentes resultados incluso cuando la imagen está comprimida o es ruidosa.

[ABC+11] presentó una técnica basada en SIFT que puede detectar regiones copiadas y pegadas. Este algoritmo también es capaz de detectar que transformación geométrica se aplicó. Debido a que la parte copiada tiene básicamente el mismo aspecto que la original, los puntos clave extraídos en la región duplicada serán bastante similares a los de la región original. El método es eficaz en imágenes comprimidas con un factor de calidad baja.

En [BJGY10] se propuso un esquema basado en los descriptores de características de velocidad de robustez (SURF), que son características de punto clave mejores que SIFT ya que funcionan mejor con técnicas de postprocesado como variaciones de brillo y borrosidad. Sin embargo, los métodos basados en puntos clave presentan un problema de salida visual debido a que las regiones copiadas y pegadas consisten en líneas y puntos que no pueden presentar un efecto visual claro e intuitivo.

3.4. Detección de Modificación o Eliminación de la Huella Digital

Gracias a la huella digital de las imágenes es posible identificar el dispositivo con el que ha sido generada dicha imagen. Identificar y garantizar la integridad del vínculo generado entre un dispositivo y su imagen es de vital importancia en el ámbito legal. Este vínculo es una evidencia válida ante un tribunal con la que posteriormente se podría llegar a determinar que individuo es propietario del dispositivo que generó la imagen.

Las técnicas forenses que hacen uso de la huella digital o ruido del sensor son esenciales para determinar el origen, la veracidad y la naturaleza de los datos. Pero también pueden usarse para llevar a cabo otras funciones de investigación multimedia. Por ejemplo, se puede determinar si una imagen es una copia de otra al buscar coincidencias en la huello digital de ambas imágenes. Las técnicas basadas en el ruido del sensor se dividen principalmente en dos categorías: defectos de píxeles y ruido del patrón del sensor (SPN).

La primera técnica estudia los defectos producidos en los píxeles, como los píxeles calientes o muertos, los defectos de filas o columnas y los defectos del grupo.

[LFG06] propone un método basado en la no uniformidad de los píxeles (PNU), que es una gran fuente para la obtención de patrones de ruido, con este método se permite la identificación de sensores y, por lo tanto, la cámara que originó la imagen. Se realizan experimentos con aproximadamente 320 imágenes tomadas con nueve cámaras digitales para estimar las tasas de acierto. Se observa que el método propuesto fue exitoso para distinguir entre dos cámaras de la misma marca y modelo. Sin embargo, los resultados obtenidos para imágenes recortadas o de diferentes tamaños no fueron satisfactorios.

En [SOAGRC+14] se estudian investigaciones recientes en el campo y proponen la mezcla de dos técnicas (imperfecciones del sensor y transformadas wavelet) para obtener una mejor identificación de fuentes de imágenes generadas con dispositivos móviles. Los resultados muestran que las imperfecciones del sensor y las transformadas wavelet pueden servir conjuntamente como buenas características forenses para ayudar a rastrear la cámara fuente de las imágenes producidas por

teléfonos móviles. Además, este modelo también permite determinar con gran precisión la marca y el modelo del dispositivo.

En [RCAGSO+13] se preseleccionó un método para la identificación de la fuente de imágenes basado en la extracción de características de ruido de foto respuesta no uniforme (PRNU), se utilizó una máquina SVM para la clasificación. Este trabajo se utilizó únicamente en dispositivos móviles y se consiguió mostrar que este método conseguía buenos resultados cuando se tenía una gran cantidad de cámaras fuente para la clasificación.

Como se puede observar la mayoría de técnicas emplean los patrones de ruido para identificar y extraer el ruido del sensor. Para poder comprobar si se ha llevado a cabo una modificación o eliminación bastaría con comparar la huella digital de la imagen original y de la imagen manipulada, esto permitiría ver diferencias entre ambas huellas digitales y así poder determinar si se ha producido una modificación.

A continuación se muestran dos tablas comparativas a modo de resumen de todas las técnicas que se han presentado a lo largo de este capítulo. Las tablas han sido divididas en técnicas con entrenamiento y técnicas sin entrenamiento.

Tabla 3.1: Comparativa de técnicas de detección de manipulaciones con entrenamiento

Enfoque	Cita	Características utilizadas	Clasificación	Datasets usados	Precisión
Retoque	[KF11]	Modificaciones geométricas y fotométricas	SVM	Sitios webs	98,75 %
Retoque	[BSVB16]	Características faciales	Red neuronal y SVM	ND-IIITD retouched faces	87 %
Retoque	[CDR13]	Características faciales	SVM	YMU, MIW	93 %
Retoque	[KAD15]	Características faciales	SVM	YMU, MIW	98,5 %
Empalme	[SCC07]	MBDCT	SVM	Columbia	91,40 %
Empalme	[ZKR08]	MBDCT, IQM	SVM	Columbia	87,10 %
Empalme	[ZZ12]	LBP, MBDCT, PCA	SVM	Columbia	89,93 %
Empalme	[WDT10]	Canales cromáticos y cadena Markov	SVM	CASIA TIDE v2.0	95,6 %
Empalme	[ZL11]	Canales cromáticos y vectores RLRN	SVM	CASIA TIDE v1.0	94,7 %
Empalme	[XYS+16]	LBP, DWT	SVM	LiveSet	92 %
Empalme	[AH17]	LBP, DCT	SVM	CASIA TIDE v2.0	97,50-97,77 %

Tabla 3.2: Comparativa de técnicas de detección de manipulaciones copia-pega

Cita	Método utilizado	Observaciones
[FSL03]	Coeficientes DCT y clasificación lexicográfica	Robusto ante retoques en la imagen
[CPF04]	Se aplica PCA, valores *Eigen* y clasificación lexicográfica	Buenos resultados ante imágenes comprimidas o ruidosas
[KW08]	SVD y clasificación lexicográfica	Validez ante filtros de desenfoque, ruido y compresión
[HGZ08]	Algoritmo de cálculo SIFT utilizando la función de coincidencia de bloques	Buenos resultados ante imágenes comprimidas o ruidosas
[HGZ08]	Algoritmo de cálculo SIFT utilizando la función de coincidencia de bloques	Buenos resultados ante imágenes comprimidas o ruidosas
[ABC+11]	Extracción de puntos clave con algoritmo SIFT	Eficaz en imágenes comprimidas con un factor de calidad baja
[BJGY10]	Basado en los descriptores de características SURF	funciona bien con técnicas de postprocesado como variaciones de brillo y borrosidad

Capítulo 4

Contribuciones

El objetivo de este capítulo es presentar las contribuciones de este trabajo. Primero se presentan un algoritmo que comprueba de forma precisa y eficaz la integridad de una imagen. A continuación, se presenta un algoritmo para identificar la región exacta duplicada de una imagen cuando se ha aplicado una técnica de manipulación copia-pega. Este algoritmo marca la zona falsificada con gran exactitud aunque la imagen haya sido modificada usando técnicas de post-procesado o recompresión.

4.1. Conceptos Generales

Las técnicas forense de detección de manipulaciones se han ido perfeccionado para hacer frente a los numerosos métodos de falsificación y manipulación de imágenes con los que se cuenta hoy en día. Es importante conocer los conceptos sobre los que se basan los algoritmos usados en dichas técnicas, a lo largo de esta sección se explicarán estos conceptos así como otros de carácter más general.

4.1.1. Modelos de Color

Los colores de una imagen se pueden representar de múltiples formas.A cada una de estas representaciones se le denomina modelo de color. Un modelo de color asigna valores numéricos a diferentes componentes de color de una imagen. Entre los más

comunes encontramos los modelos RGB, CMYK y YCbCr.

- **El modelo RGB**: Es uno de los más conocidos debido a su uso extendido en ordenadores, pantallas, televisores y dispositivos móviles, entre otros. Se basa en la representación de un color mediante la mezcla por adición de los colores rojo, verde y azul. Este modelo asigna valores entre 0 y 255 a cada uno de sus tres componentes (rojo, verde y azul), la suma de las tres formará el valor que representa el color. Una imagen digital está formada por multitud de píxeles, cada uno de ellos tendrá su propio valor dentro del modelo RGB siendo 0 el color negro y 255 el color blanco.
- **El modelo CMYK**: Este modelo se basa en representar un color en los componentes cian, magenta, amarillo y negro. Esta representación al contrario que el modelo anterior se hace de manera sustractiva y no por adición, es decir, mezclando sus tres componentes cian, magenta y amarillo se consigue un tono muy similar al color negro y no a el color blanco como pasaba en RGB. Es por esto que necesita de una cuarta componente para representar el tono de negro adecuado. Los valores que asigna a cada componente van desde 0 al 100 en función de la intensidad que se le quiera dar al tono del componente. Este modelo es muy usado en el sector de la imprenta debido al buen contraste con el que genera los diferentes tonos de las impresiones. Por tanto, antes de la impresión de una imagen se suele transformar el modelo RGB al modelo CMYK con algún tipo de software de edición de fotografías.
- **El modelo YCbCr**: Es un modelo de color usado para el procesamiento de imágenes digitales, este modelo se trata de una codificación no lineal del espacio RGB. Representa los colores en forma de componentes de luminancia y de crominancia. La luminancia se representa con el símbolo Y y las dos señales de crominancia se representan con Cb y Cr. Cb sitúa el color en una escala comprendida entre el color azul y amarillo, en cambio Cr lo hace entre el color rojo y verde. Por último, el parámetro Y ofrece la información respectiva a los colores blanco y negro.

  En la figura 4.1 puede observarse una imagen y sus componentes del modelo $YCbCr$ en el siguiente orden: luminancia, canal cromático azul y canal cromático rojo.

Figura 4.1: Imagen en el modelo de color *YCbCr*

4.1.2. Patrón Binario Local

LBP es un operador local que discrimina diferentes tipos de texturas. Cuando se realiza una manipulación, la textura original de la imagen se distorsiona.

El operador LBP define una etiqueta denominada código LBP para cada píxel de una imagen. Si el valor del píxel vecino es menor que el del píxel central, se asignará a dicho vecino el dígito binario 0, de lo contrario se le asignará el dígito binario 1. Una vez se obtienen los valores de la matriz en representación de 0 y 1 se convierte el número binario en un número decimal en el sentido de las agujas del reloj, de esta manera se obtiene el código LBP que representa al píxel central. Cuando se trabaja con LBP se asigna un valor de vecindad representado por la letra P. Este valor hace referencia al número de vecinos que rodean al píxel central. En [AHA+13] se observó que con el valor de vecindad establecido en 8 se obtiene una mayor precisión en los resultados finales. En la Figura 4.2 se observa una imagen manipulada y su resultado al aplicar LBP. La imagen ha sido obtenida del dataset CASIA v1.0.

Figura 4.2: Imagen manipulada y su transformación al aplicar LBP

4.1.3. Histograma

Un histograma es una representación de los datos donde estos son agrupados por su frecuencia de aparición, es muy usado en las técnicas de detección de falsificaciones debido a su eficacia para reducir el número de datos con los que se está trabajando. Cuando se extraen características de una imagen es necesario realizar un procesamiento previo de reducción ya que no es eficiente trabajar de forma directa con el número total de características extraídas.

4.1.4. Transformada Discreta del Coseno

La transformada discreta del coseno (DCT) es una variación de la transformada discreta de Fourier, donde la imagen se descompone en sumas de cosenos, pero utilizando números reales únicamente. Sólo actúa sobre funciones periódicas con simetría par y el resultado es una secuencia de números reales. En la ecuación 4.1 se muestra la fórmula característica de la variante DCT-I.

$$f_j = \frac{1}{2}(x_0 + (-1)^j x_{n-1}) + \sum_{k=1}^{n-2} x_k \cos\left[\frac{\pi}{n-1} kj\right] \tag{4.1}$$

DCT es usada en detección de manipulaciones y sobre todo en algoritmos de compresión como JPEG por su gran capacidad de compactar la energía o la mayor parte de información en un número reducido de coeficientes y por ser independiente del número de datos de entrada que recibe garantizando una mayor eficiencia al trabajar con imágenes de grandes dimensiones.

DCT consigue llevar a cabo la compresión de imágenes descartando las partes imperceptibles para el ojo humano. Este proceso usa los coeficientes DCT para diferenciar que puntos de la imagen presentan características diferentes al resto o cuales son similares. Es por esto que es muy usado en la detección de manipulaciones en imágenes, trabajar con DCT en el canal de crominancia va a permitir obtener los coeficientes que indicarán puntos de la imagen que sobresalen del resto pero que a simple vista no son perceptibles.

Existen ocho tipos de DCT, la más utilizada en la compresión de imágenes

digitales y por lo tanto en el ámbito de la detección de falsificaciones es la DCT-II (ecuación 4.2) cuya inversa DCT corresponde al tipo DCT-III mostrado en la ecuación 4.3.

$$f_j = \sum_{k=0}^{n-1} x_k \cos\left[\frac{\pi}{n} j \left(k + \frac{1}{2}\right)\right] \tag{4.2}$$

$$f_j = \frac{1}{2} x_0 + \sum_{k=1}^{n-1} x_k \cos\left[\frac{\pi}{n} \left(j + \frac{1}{2}\right) k\right] \tag{4.3}$$

4.1.5. Transformada Discreta Wavelet

Una función wavelet es una pequeña onda que concentra su energía en el tiempo, se usa para analizar en términos de tiempo y frecuencia fenómenos estacionarios, no estacionarios y variables de tiempo. La transformada wavelet es considerada como la forma de representación de señales más potente que puede aplicarse principalmente al procesamiento de señales y de imágenes. Existen numerosos tipos de familias de wavelets, las más importantes son: Haar, Daubechies, Coiflets, Symlets, Biortogonales, Meyer, Mexican hat, Shannon y Morlet.

La transformada discreta wavelet (DWT) puede proporcionar representaciones únicas y discriminatorias que pueden cuantificar estructuras vitales e interesantes como bordes y detalles con buena resolución por pocos coeficientes. También es computacionalmente efectiva debido al número reducido de datos con el que trabaja. Los coeficientes wavelets finales se pueden usar directamente como características y pueden ser extraídas directamente del dominio wavelet, describiendo las anomalías en los datos de la imagen. Básicamente la transformada wavelet discreta reduce la correlación entre los coeficientes ondulatorios y proporciona compactación de energía en algunos coeficientes wavelet.

4.1.6. Máquina de Soporte Vectorial

Es una técnica supervisada de aprendizaje automático útil para resolver problemas de reconocimiento de patrones y análisis de regresión. El objetivo de SVM es encontrar el mejor hiperplano que divide los datos en dos clases diferentes.

El paquete de software LIBSVM es la implementación de SVM más utilizada. Cuando se usa SVM surgen dos problemas:

- El primero está relacionado con la selección de la función del kernel. De acuerdo con la lineal separable y lineal inseparable, se puede utilizar diferentes funciones del núcleo. Para hacer la clasificación de las muestras más fácil y más precisa se utiliza el núcleo de la función de base radial (RBF), el cual hace un mapeo no lineal a un espacio de alta dimensión.

- El segundo problema es la selección de los parámetros apropiados del kernel. Hay dos parámetros en la función RBF del kernel (C y γ). Para encontrar los mejores parámetros de clasificación de prueba y entrenamiento se utiliza el método de optimización de parámetros. El archivo ejecutable del método de optimización de parámetros en LIBSVM es *gunplot.exe*.

4.2. Consideraciones Generales

Los conceptos presentados en la sección 4.1 pueden ser utilizados como características de una imagen. Dichas características aportan información que puede ser utilizada para poder detectar las distintas manipulaciones en la imagen. Este trabajo combina dichas características. De esta forma el proceso de extracción de las características no se realiza de forma independiente si no que se procesan de forma conjunta. Así las características finales son la aplicación de distintos métodos combinados.

Los dos algoritmos de identificación de manipulaciones basados en entrenamiento hacen uso de distintos métodos; El primero se basa en la división de la imagen en bloques superpuestos y realiza una combinación de histograma aplicando LBP y DCT. El segundo algoritmo, se basa en el uso de coeficientes DWT, la aplicación de QMF y la obtención del histograma LBP.

Finalmente, el algoritmo de detección de la región duplicada usando la técnica copia-pega se basa en la división superpuesta de bloques y en la extracción de coeficientes DCT.

4.3. Algoritmos de Identificación de Manipulaciones Basado en Entrenamiento

En esta sección se presentan los algoritmos basados en entrenamiento y clasificación.

- Entrada: Imagen de la que se extraerán las características.
- Salida: Vector de características que se usará para el entrenamiento y clasificación.

4.3.1. Algoritmo Basado en la Transformada Discreta del Coseno

Cada imagen MxN se convierte al modelo de color $YCbCr$. Como la visión humana percibe el componente de luminancia de una manera más clara que el componente de crominancia. Se considera que, la mayoría de las trazas de manipulación, que no pueden detectarse a simple vista, pueden estar en el canal cromático. Este canal describe el contenido de la señal de la imagen, como los bordes. Cualquier inconsistencia en estos bordes causada por la operación de una alteración se enfatiza y, por tanto, se nota. El componente Y se descarta y se hace uso solamente de los dos componentes de crominancia.

Cada componente de la crominancia es dividido en bloques superpuestos de 16x16.

Debido a la capacidad de LBP para capturar las diferencias de textura, es una herramienta muy eficiente para la detección de las falsificaciones en las imágenes digitales. Se extrae por cada bloque el patrón binario local utilizando como valor de vecindad 8 para obtener una mayor precisión. Para reducir el número de características obtenidas con LBP se calcula el histograma, quedando así cada bloque representado por 256 códigos LBP.

De esta manera se observan los cambios de las diferentes texturas con una mayor eficiencia. En este caso las características de textura locales de toda la imagen se

pueden describir mediante un histograma normalizado que está formado por 2^P códigos LBP, siendo P el número de vecinos que rodean al píxel central. Para el caso de $P = 8$ el número de características totales será de 256 códigos LBP por cada imagen.

Una vez obtenidos los histogramas de todos los bloques de cada componente se aplica la transformada discreta del coseno a cada uno de ellos utilizando la Fórmula 4.2. Con esto se representa cada bloque por un conjunto de coeficientes DCT. El resultado de aplicar la transformada discreta del coseno es la obtención de una secuencia finita de puntos como resultado de la suma de varias señales con distintas frecuencias y amplitudes. De cada conjunto de coeficientes se obtiene la desviación estándar como característica.

Finalmente, cada componente de crominancia quedará representado como un vector de desviaciones estándar de los conjuntos de coeficientes DCT de todos sus bloques. El vector final de características que se enviará al clasificador SVM será la concatenación de los vectores de ambas crominancias. Para encontrar los mejores pares de parámetros óptimos de clasificación C y γ se utilizará la herramienta *Grid-search* y *Cross-validation*.

En la Figura 4.3 se muestra un diagrama del proceso de extracción de características que realiza este algoritmo.

4.3.2. Algoritmo Basado en la Transformada Discreta Wavelet

Cada imagen original MxN se convierte al modelo de color $YCbCr$. Como la visión humana percibe el componente de luminancia de una manera más clara que el componente de crominancia. Se considera que, la mayoría de las trazas de manipulación, que no pueden detectarse a simple vista, pueden estar en el canal cromático. Este canal describe el contenido de la señal de la imagen, como los bordes. Cualquier inconsistencia en estos bordes causada por la operación de una alteración se enfatiza y, por tanto, se nota. El componente Y se descarta y se hará uso solamente de los dos componentes de crominancia (Cb y Cr).

Por cada componente Cb y Cr de la imagen se aplica DWT. La transformada

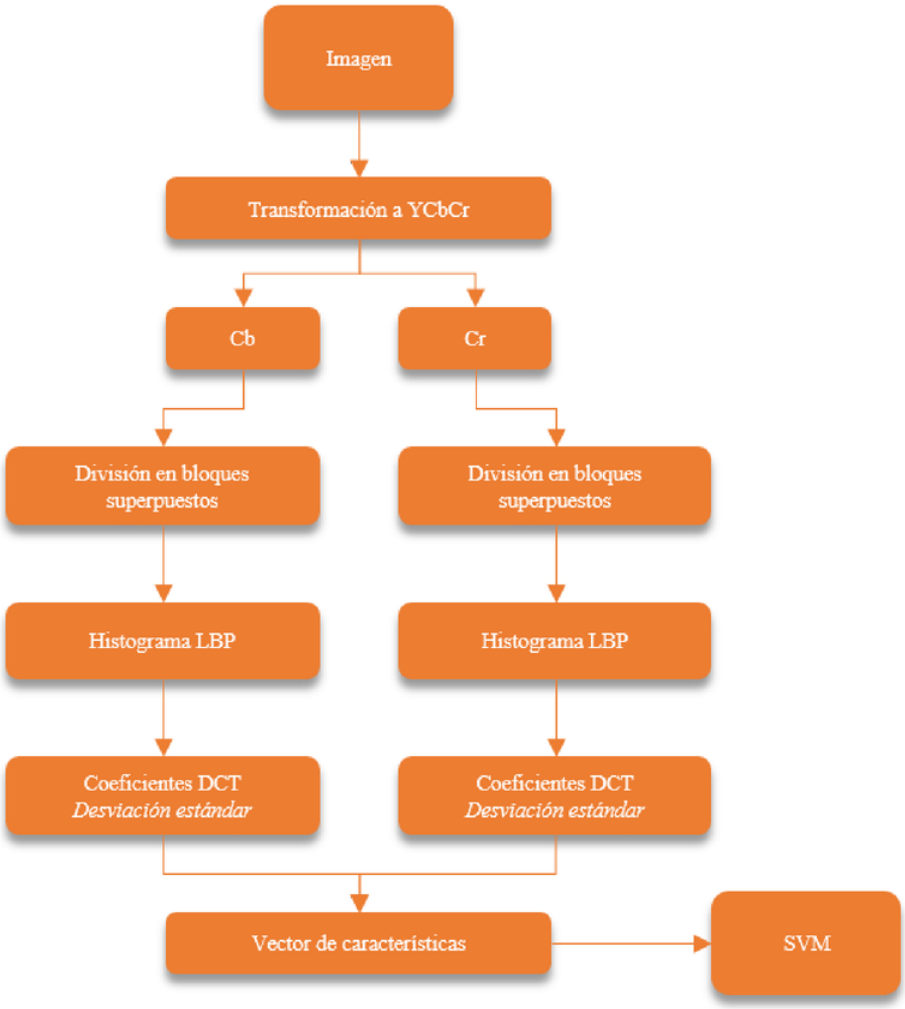


Figura 4.3: Diagrama del algoritmo basado en DCT

discreta wavelet analiza una imagen en diferentes escalas y orientaciones. El proceso de empalme a menudo introduce una transición nítida en la matriz bidimensional que representa a la imagen en términos de bordes, líneas y esquinas que se caracterizan por componentes de alta frecuencia en el dominio de transformada de Fourier.

Cada componente Cb y Cr quedará descompuesto por cuatro sub-imágenes: el coeficiente de aproximación, coeficiente horizontal, coeficiente vertical y coeficiente diagonal, respectivamente. El tamaño de cada coeficiente es $\frac{1}{4}$ del tamaño de la

imagen que se está procesando.

Entre los cuatro coeficientes, el coeficiente de aproximación (LL) es similar a la imagen. Sin embargo, los tres coeficientes de dirección restantes (HL, LH y HH) contienen alguna información de alta frecuencia en las diferentes direcciones.

Seguidamente, a cada componente wavelet (HL, LH, HH y LL) se aplica QMF o filtro espejo en cuadratura para obtener la codificación inversa de las frecuencias, es decir, las frecuencias altas se intercambian por las bajas y viceversa

A continuación, se aplica LBP a cada componente wavelet utilizando como valor de vecindad 8 para obtener una mayor precisión. Dado que la imagen de codificación LBP incluye información de modo micro local de la imagen original, las características locales de la imagen se pueden describir a través de un histograma que está formado por 256 códigos LBP.

Una vez obtenido los histogramas para cada una de las sub-imágenes o coeficientes DWT, se procede a la creación del vector de características que será enviado a la máquina de soporte vectorial.

Debido a que las imágenes se han descompuesto en cuatro coeficientes diferentes, el vector de características se construye calculando 256x4 características de textura por cada uno de los canales de crominancia (Cb y Cr).

En la figura 4.4 se muestra un diagrama que indica los diferentes pasos que realiza el algoritmo.

4.4. Algoritmo de Identificación de la Región Exacta Duplicada en Técnicas Copia-Pega

En esta sección se propone un esquema mejorado de detección de falsificación de copia-pega basado en el esquema presentado por primera vez por Fridrich [FSL03]. A continuación se especifican los parámetros de entrada y los resultados que genera el algoritmo tras su ejecución.

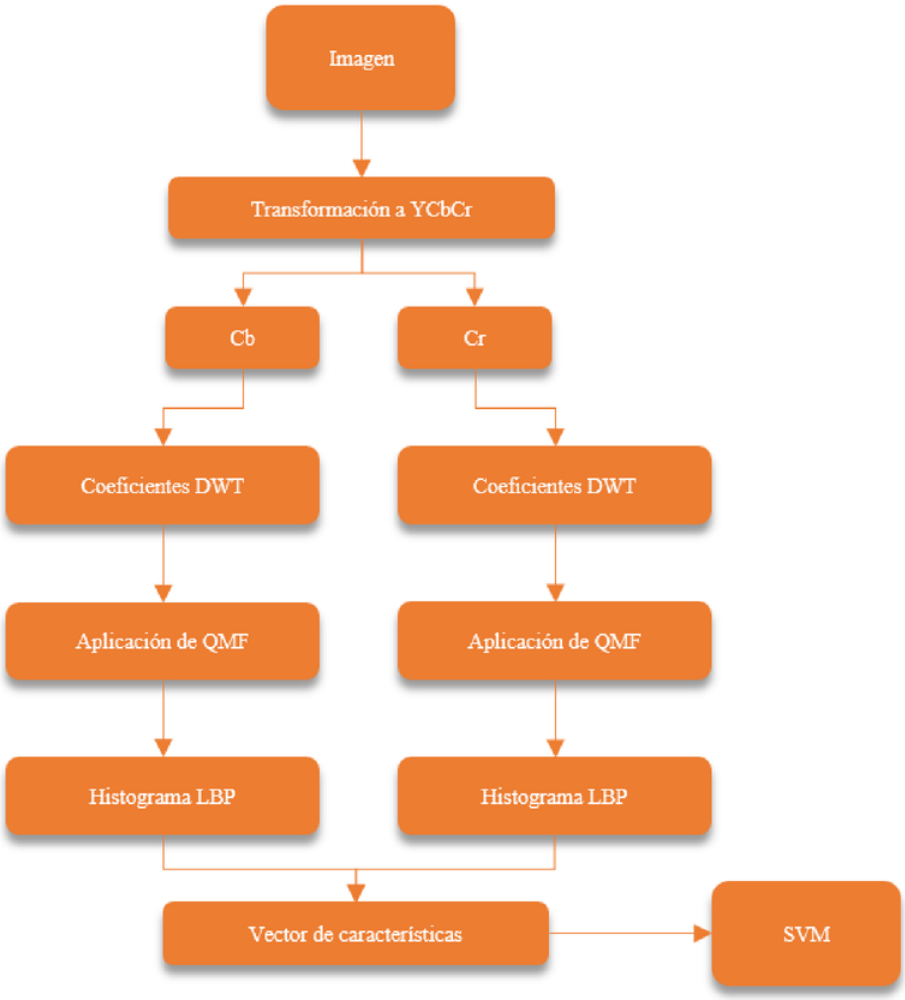


Figura 4.4: Diagrama del algoritmo basado en DWT

- Entrada: Imagen a analizar.
- Salida: Imagen con la región duplicada marcada de un color determinado, de esta forma se puede visualizar claramente el área sobre el que se ha realizado la copia y la región exacta donde se ha pegado.

El primer paso es convertir la imagen que se desea analizar a escala de grises. Para ello se extraen los componentes del canal de luminancia y se representa la imagen con ellos.

A continuación, se establece un tamaño de bloque $B = 8$ para dividir la imagen desde la esquina superior izquierda a la esquina inferior derecha.

Los bloques se superponen con un desplazamiento de un píxel hasta obtener $(M–B + 1)(N–B + 1)$ bloques superpuestos, siendo M y N las dimensiones de la imagen. El tamaño de bloque B se ha establecido en 8 para conseguir resultados más precisos con un nivel óptimo de ruido.

A continuación, se extrae las características DCT de cada uno de los bloques. DCT puede eliminar la redundancia entre píxeles adyacentes de manera rápida y efectiva, y tiene propiedades de compactación de energía [FZWJ16], por ello es razonable adoptar los coeficientes DCT como características de los bloques de imagen. Una propiedad de DCT es que la energía solo se enfoca en los coeficientes de baja frecuencia, es decir, no todos los elementos son igual de importantes, por ello se descartan los coeficientes de alta frecuencia por que solo introducen ruido y pueden dar lugar a errores en procesos posteriores.

Para llevar a cabo este proceso se establece un valor de truncamiento y se realiza a su vez un escaneado en zigzag. El valor de truncamiento k se calcula mediante la ecuación 4.4 y corresponde a la longitud del vector de características de un bloque. Para establecerlo se fija un factor de truncamiento $f_t (0 < f_t < 1)$.

$$k = \left[f_t . B^2 \right] \tag{4.4}$$

A su vez se realiza un escaneado en *zigzag* sobre el bloque de coeficientes DCT como se muestra en la Figura 4.5. Este tipo de escaneo permite realizar un recorrido ascendente por los coeficientes de menor a mayor frecuencia y gracias al valor k antes mencionado accede solo a los coeficientes más importantes. La zona verde de la Figura 4.5 representa el área de coeficientes DCT más significativos.

Por último, para reducir las dimensiones y mejorar la eficiencia del proceso de adaptación, los coeficientes DCT se cuantifican mediante un factor de cuantificación f_q y se redondean al entero más cercano usando la ecuación 4.5.

$$\vec{a_i} = \left(\left[\frac{a_{i1}}{f_q} \right], \left[\frac{a_{i2}}{f_q} \right], ..., \left[\frac{a_{ik}}{f_q} \right] \right) \tag{4.5}$$

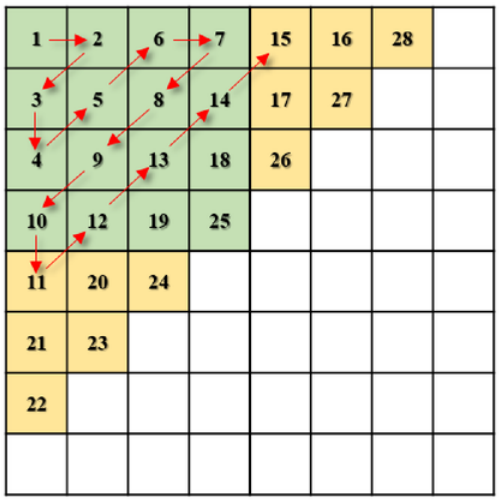


Figura 4.5: Escaneo en zig-zag

Este proceso proporciona una secuencia con los coeficientes de baja frecuencia agrupados y truncados para cada bloque de la imagen, $(M - B + 1)(N - B + 1)(f_t.B^2)$ bloques totales.

Finalmente, se crea una matriz A de una sola columna para guardar cada secuencia en una fila diferente, a la vez que se añaden las coordenadas x e y de la esquina superior izquierda del bloque al final de la secuencia de coeficientes.

El siguiente paso es ordenar lexicográficamente la matriz A con todos los vectores de características, de esta manera las filas de características similares quedarán juntas y así se podrá determinar qué bloques de la imagen están relacionados. Por lo tanto, se requieren algunos métodos para juzgar si los vectores de características correspondientes de los bloques de imagen son los mismos.

Si los componentes correspondientes de los dos vectores de bloques de imagen son casi iguales, los dos bloques de imagen pueden considerarse estrechamente relacionados. Posteriormente, estos bloques se estudiarán para determinar si uno de ellos o ambos son objetos de manipulación.

Para juzgar la similitud entre dos bloques se realizan las siguientes comprobaciones para cada fila de la matriz A:

- Cada vector de fila $\vec{a_i} = (a_i^1, a_i^2, ..., a_i^k)$ debe ser comparado con sus vectores de fila adyacentes $\vec{a_j} = (a_j^1, a_j^2, ..., a_j^k)$. Se define un parámetro N_a que corresponde al número de máximo de filas que van a ser comparadas con a_i, por lo que debe satisfacer $(j–i < N_a)$.

- A continuación, se definen los umbrales S_t y T_t que serán usados más adelante y se inicializa la variable r_{max} a un valor suficientemente pequeño y la variable r_{min} a un valor suficientemente grande. Se crea un contador c inicializado a 0; entonces se procede a los siguientes pasos: Para cada $\vec{a_i}$ y $\vec{a_j}$ dentro del intervalo $(1 <= l <= k)$:
  1. Si $a_j^l = 0$ se comprueba si se cumple $|a_i^l – a_j^l| < S_t$, en caso afirmativo se incrementa el valor de c en 1.
  2. Si $a_j^l \neq 0$ se calcula $r_i = a_i^l / a_j^l$ y se cambian los valores r_{min} y r_{max} según corresponda:
     - Si $r_{max} < r_l$ entonces $r_{max} = r_l$
     - Si $r_{min} > r_l$ entonces $r_{min} = r_l$
  3. Si se cumple que $r_{max} – r_{min} > T_t$ se incrementa c en 1.
  4. Finalmente si $c < C_t$ entonces a_i y a_j son similares.

Después de comprobar si el vector de fila a_i (la coordenada superior izquierda del bloque de imagen es (x_1, y_1)) y el vector de fila a_j (la coordenada superior izquierda del bloque de imagen es (x_2, y_2)) son similares, se calculan los vectores de transferencia entre los dos vectores, $\vec{s} = (s_1, s_2) = (x_1 - x_2, y_1 - y_2)$.

A continuación se comprueba si la distancia excede de un parámetro T_d, con la ecuación 4.6. Si se cumple, la frecuencia existente del vector de transferencia se incrementa en 1, en caso contrario, no se modifica dicha frecuencia.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} > T_d \tag{4.6}$$

Una vez obtenidas las frecuencias para los vectores de transferencia se procede a buscar el vector principal de transferencia cuyas frecuencias exceden un umbral T_f. Los bloques de la imagen correspondientes al vector principal se pueden considerar como regiones copiadas y pegadas. Estas regiones se marcan respectivamente en un color que las haga distinguirse del resto de la imagen.

La Figura 4.6 presenta el diagrama de los procesos más característicos del algoritmo.

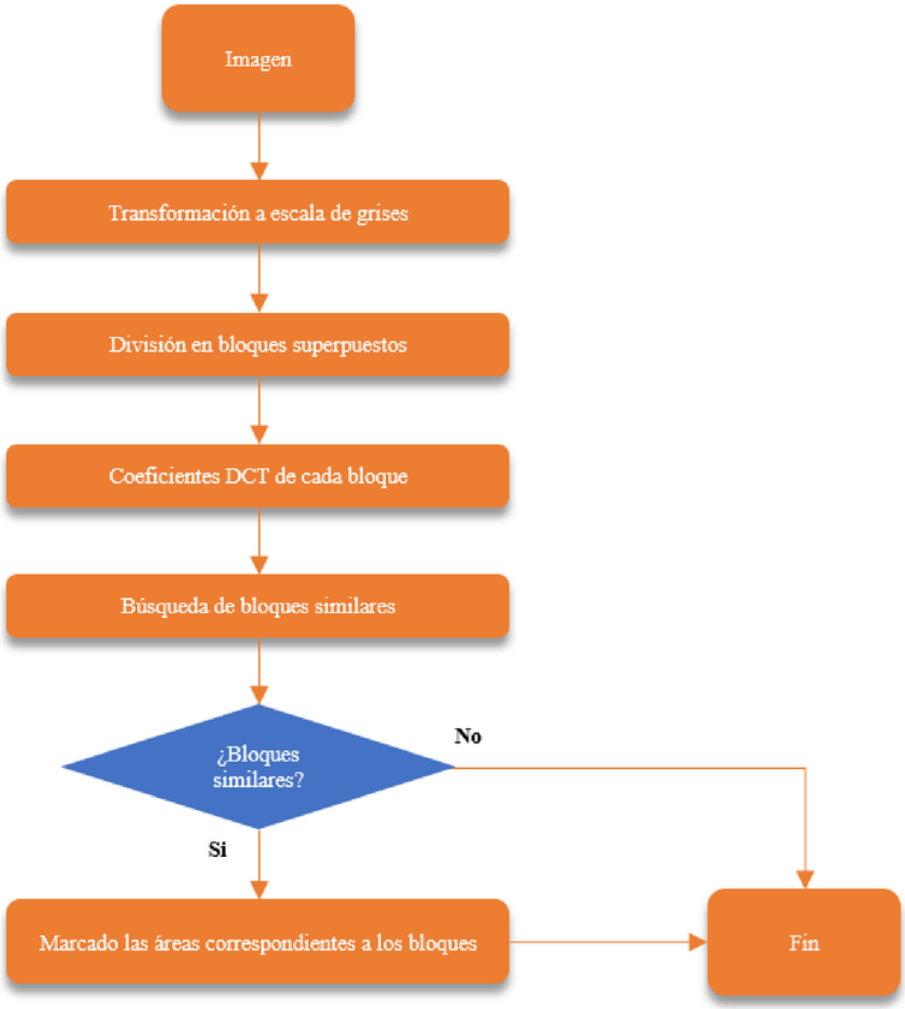


Figura 4.6: Diagrama del algoritmo de identificación de Copia-Pega

Capítulo 5

Experimentos y Resultados

En este capítulo se describen los experimentos realizados para evaluar la eficacia de los algoritmos propuestos y los resultados obtenidos. El capítulo está dividido en 3 secciones. La sección 5.1 presenta las configuraciones de los experimentos realizados. La sección 5.2 está enfocada a la evaluación de los algoritmos de identificación de manipulaciones basado en entrenamiento, se recogerán experimentos realizados tanto para el algoritmo basado en DCT, como para el basado en DWT. El segundo grupo de experimentos se hace sobre el algoritmo de detección de falsificaciones basado en técnicas copia-pega (sección 5.3).

5.1. Configuración de los Experimentos

En todos los experimentos realizados se ha utilizado *Python* como lenguaje de programación, debido a su gran flexibilidad para poder realizar el análisis de datos y su alta velocidad en gestionar la entrada y salida.

La detección de manipulaciones es un problema de dos clases, imágenes auténticas contra manipuladas. Es por ello que para poder llevar a cabo el entrenamiento y clasificación de los datos se ha utilizado una máquina de soporte vectorial (SVM), ya que tiene un rendimiento excelente al trabajar con problemas de dos clases. Se ha usado el paquete de software LIBSVM [CL11], debido a su simplicidad y eficacia. Esta implementación de SVM es a día de hoy una de las

herramientas más comúnmente utilizadas. Como función del núcleo se usa la función de base radial (RBF), esta función es la más utilizada en proyectos similares y ha demostrado muy buenos resultados. Los valores óptimos de los parámetros de la función del núcleo RBF (γ, C) se calculan automáticamente mediante un proceso de correlación cruzada.

Para la evaluación de estos algoritmos se ha hecho uso de varios datasets públicos ([DW], [IT14]) para realizar experimentos con varios formatos y tamaños. La Tabla 5.1 muestra las características de los datasets utilizados en los experimentos.

Tabla 5.1: Características de los datasets utilizados

Datasets	Formato	Resolución	Número de Imágenes		
			Originales	Manipuladas	Total
CASIA v1.0 [DW]	JPEG	384x256	800	921	1721
CASIA v2.0 [DW]	JPEG,BMP,TIFF	240x160,900x600	7491	5123	12614
IFS-TC [IT14]	PNG	1024x768,3648x2736	424	451	875

Las características del equipo en el cual se han realizado los experimentos se presentan en la Tabla 5.2. Es un factor importante a tener en cuenta ya que los tiempos de ejecución de las diferentes pruebas varían según los recursos computaciones disponibles.

Tabla 5.2: Características del equipo de experimentación

Recursos	Características
Sistema operativo	Ubuntu 17.04
Memoria	12 GB
Procesador	Intel® Core™ i5-6200U CPU @ 2.30GHz x 4
Gráficos	Intel® HD Graphics 520 (Skylake GT2)
Tipo de SO	64 bits
Disco	64 GB

5.2. Evaluación de los Algoritmos de Identificación de Manipulaciones Basado en Entrenamiento

A lo largo de esta sección se mostrarán todos los experimentos que se han realizado para evaluar la efectividad de los algoritmos de identificación de manipulaciones basado en entrenamiento.

5.2.1. Experimento 1

Este experimento se basó en comprobar la variación de la precisión al aplicar QMF sobre el algoritmo de identificación de manipulaciones basado en DWT. Se realizo una prueba aplicando LBP sobre los coeficientes wavelets de manera directa y otra pasando los coeficientes waveletes por QMF antes de aplicar LBP. Para ello se usaron los datasets CASIA v1.0 y CASIA v2.0. En la Tabla 5.3 pueden verse los resultados obtenidos.

Tabla 5.3: Variación de las precisiones al aplicar QMF

Datasets	DWT-LBP-HIST	DWT-QMF-LBP-HIST
CASIA v1.0	97.66 %	98.01 %
CASIA v2.0	98.73 %	99.43 %

Como se puede observar en la tabla, el algoritmo que hace uso de QMF consigue un leve aumento de la precisión en ambos datasets.

5.2.2. Experimento 2

En este experimento se buscó cual de los dos algoritmos de detección de manipulaciones propuestos presentaba mayor precisión y eficiencia. En la Tabla 5.4 se muestran los resultados obtenidos.

- Se observó que el algoritmo basado en DCT presenta más retardo que el

algoritmo basado en DWT. Esto se debe a la división en bloques superpuesta que realiza. Como referencia se midió el tiempo de ejecución de ambos algoritmos para una imagen de dimensiones 1280x854. El algoritmo basado en DCT presentaba un tiempo de ejecución de 89 segundos, siendo más eficiente el algoritmo basado en DWT con un tiempo de ejecución de 16 segundos.

- También se observo que a mayor tamaño de imagen el algoritmo basado en DCT incrementa considerablemente el tiempo de procesado de las imágenes. En cambio en el algoritmo basado en DWT se produce un incremento leve del tiempo de ejecución.
- Finalmente, ambos algoritmos consiguen buenos resultados con imágenes comprimidas, como por ejemplo imágenes con formato JPEG. También se obtuvieron buenos resultados para imágenes con formato PNG. Los dos algoritmos no consiguen llegar a una precisión superior al 50 % en imágenes con formato TIFF.

Tabla 5.4: Precisión obtenida por ambos algoritmos

Datasets	DWT-QMF-LBP-HIST	LBP–HIST-DCT
CASIA v1.0	96,57 %	53,72 %
CASIA v2.0	99,43 %	94,94 %
IFS-TC	97,56 %	70,22 %

Como se puede observar en la tabla el algoritmo basado en DWT obtiene mejores resultados en los tres datasets y como sé ha mencionado anteriormente con un tiempo de procesado por imagen inferior al algoritmo basado en DCT.

5.3. Evaluación del Algoritmo de Identificación de la Región Exacta Duplicada en Técnicas Copia-Pega

En esta sección se mostrarán los experimentos que se han realizado para evaluar la efectividad del algoritmo de identificación de la región exacta duplicada en técnicas copia-pega. A lo largo de las pruebas realizadas se ha podido comprobar que el algoritmo funciona con cualquier tipo de formato, como JPEG, PNG, BMP, entre

otros. También hay que destacar que el tamaño de la imagen no influye en la precisión de los resultados, solo produce variaciones en el tiempo de procesamiento como se muestra en el Experimento 3.

5.3.1. Experimento 1

En este experimento se comprobó la efectividad del algoritmo propuesto en el capítulo 4, sección 4.4. Este algoritmo hace uso de diferentes parámetros configurables, dependiendo del valor asignado los resultados pueden variar notablemente. En [ZWWZ17] se propone un algoritmo que da excelentes resultados en identificación de manipulaciones copia-pega. Para realizar sus experimentos hacen comparaciones entre los parámetros usados por otras investigaciones. Los valores que han establecido han servido como referencia para inicializar los parámetros del algoritmo que se ha propuesto en este trabajo. En la siguiente tabla se exponen cada uno de los parámetros utilizados y sus valores correspondientes.

Tabla 5.5: Parámetros configurables del algoritmo copia-pega

Parámetro	Nombre	Valor asignado
ft	Factor de truncamiento	0,25
fq	Factor de cuantificación	4
Na	Filas vecinas comparables	3
St	Umbral S	4
Tt	Umbral T	0,06
Ct	Umbral de similitud	3
Tf	Umbral de frecuencia	50
Td	Distancia de los vectores	20

El parámetro que mejora los resultados ha sido el *umbral de frecuencia* o *Tf*. Este parámetro establece el valor con el cual un bloque de la imagen puede considerarse una manipulación válida. Si un bloque aparece varias veces en la imagen como duplicado y dicha frecuencia de aparición supera a la establecida por el umbral *Tf* se considerará que forma parte de la manipulación. Estudiar la frecuencia de aparición de los bloques es posible gracias a la superposición con la que se extraen de la imagen. Cuando este parámetro es alto los resultados finales son más refinados, eliminando las áreas identificadas como manipuladas pero que en realidad son falsos positivos. En el experimento se ha ajustado el parámetro *Tf* en tres valores: 50, 100 y 150.

En la Figura 5.1 se muestran los resultados de la detección con diferentes valores del parámetro *Tf*.

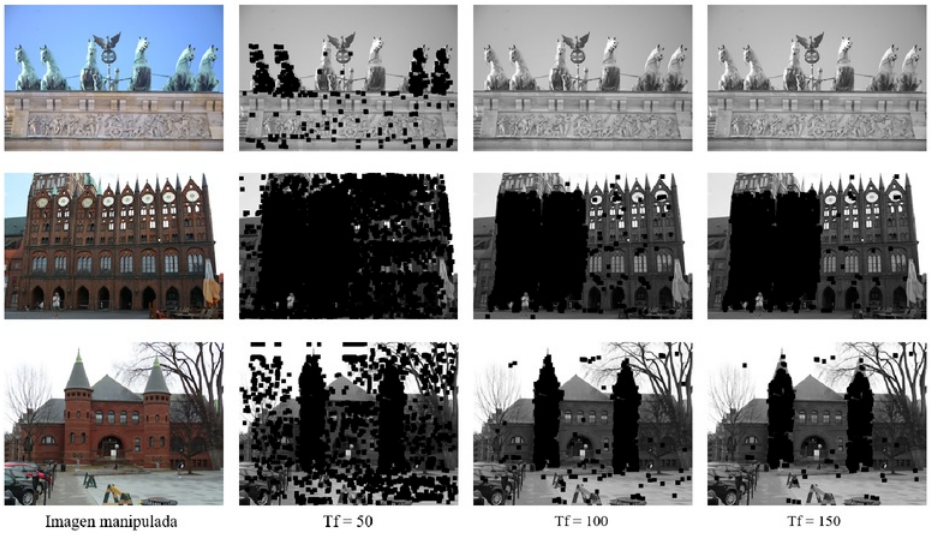


Figura 5.1: Identificación de la manipulación copia-pega

Como se puede observar en la figura a mayor valor del parámetro *Tf* los resultados presentan menos ruido, es decir, se eliminan las zonas negras que no forman parte de la manipulación. En la primera imagen la manipulación se identifica con el *Tf*=50, con un valor más alto el algoritmo no encuentra ningún bloque duplicado que cumpla la frecuencia de aparición establecida por *Tf*. En cambio puede apreciarse en las otras dos imágenes que a mayor valor del parámetro *Tf* se elimina el ruido producido por los falsos positivos. Esto se debe a que son manipulaciones de gran tamaño en proporción a la imagen por lo que la frecuencia de aparición de los bloques manipulados será muy superior al existir la superposición.

Sin embargo, el algoritmo falla con un tipo concreto de manipulaciones. Estas manipulaciones consisten en duplicar determinadas áreas que presentan zonas de la imagen real. Esto se puede observar en el ejemplo de la Figura 5.2.

En esta imagen se ha duplicado el árbol situado en la parte central. Este árbol presenta huecos entre las ramas que han sido editados en la duplicación para que se integre perfectamente con el fondo, es por ello que el algoritmo trata ambos árboles como objetos diferentes y no es capaz de dar un resultado correcto.

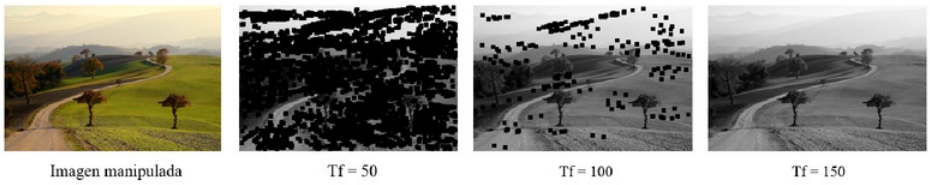


Figura 5.2: Área duplicada con detalles de la imagen real

5.3.2. Experimento 2

En el segundo experimento se comprobó la precisión del algoritmo de identificación de la región de copia-pega en imágenes de texturas con patrones similares. En este tipo de imágenes la manipulación pasa inadvertida debido a su excelente integración con el fondo original. Esto se debe a que se usa un mismo patrón de colores sin áreas que resalten por encima de otras. En este experimento se realizaron dos pruebas con este tipo de imágenes, se ajustó el parámetro Tf al valor 150 para disminuir el ruido de puntos negros en los resultados.

En la primera prueba se usaron imágenes con muchos detalles de múltiples colores pero a su vez siguen un mismo patrón, esto hace que el área duplicada sea difícil de detectar. En la Figura 5.3 se muestran tres ejemplos de identificación en este tipo de imágenes. Como se observa, el algoritmo consigue una precisión destacable.

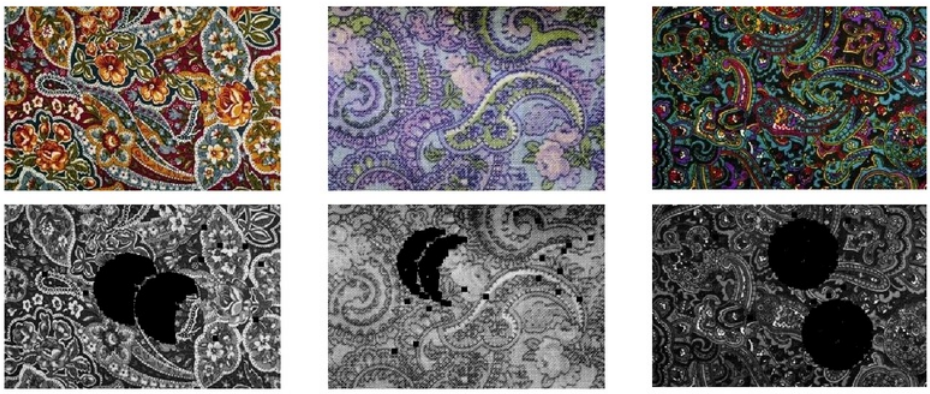


Figura 5.3: Identificación de copia-pega en imágenes de texturas con patrones similares

Para la segunda prueba se usaron imágenes donde la región duplicada se encontraba en un área del mismo color que otras regiones de la imagen. En este tipo de imágenes también es difícil detectar la región duplicada ya que puede confundirse con otra región original que tenga el mismo color. En la Figura 5.4 se muestran tres ejemplos donde puede apreciar que el algoritmo presenta un buen funcionamiento ante este tipo de manipulaciones.

Figura 5.4: Identificación de copia-pega en imágenes con áreas del mismo color

5.3.3. Experimento 3

En este experimento se analizó la eficiencia del algoritmo en imágenes de gran tamaño y resolución. Se observó que al escalar una imagen a un tamaño más pequeño la precisión del algoritmo sigue manteniéndose alta sin sufrir cambios significativos. Esta observación permite realizar un escalado de las imágenes grandes antes de que el algoritmo las procese, esto aumenta la eficiencia sin perder calidad en los resultados.

En la Figura 5.5 se muestra un ejemplo de una imagen manipulada por la técnica copia-pega, en la cual se ha copiado el pájaro situado encima del césped y se ha pegado sobre la cabeza de la vaca. El tamaño original de la imagen es de 1080x854 píxeles, también se muestra el resultado obtenido al escalar la imagen a un tamaño de 640x427 píxeles. El tiempo de ejecución que ha tardado el algoritmo en procesar la imagen original ha sido de 160 segundos, en cambio en la imagen escalada ha tardado 48 segundos. Como puede observarse se ha detectado la manipulación perfectamente en ambas imágenes, por lo que es posible realizar el escalado sin afectar a la precisión

del algoritmo y mejorando considerablemente el tiempo de ejecución.

Figura 5.5: Identificación de copia-pega en imágenes escaladas

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

En la actualidad nos encontramos con numerosas aplicaciones que consiguen realizar ediciones en imágenes con resultados altamente profesionales. Detectar si una imagen ha sido alterada mediante alguna técnica de manipulación es una tarea complicada en la que hoy en día se sigue investigando para encontrar técnicas forenses que puedan detectar estas falsificaciones.

En este trabajo se ha realizado un estudio exhaustivo sobre las técnicas más importantes para la identificación de manipulaciones enfocándose más en la identificación de empalme y manipulaciones copia-pega. Tras comparar las diferentes investigaciones se ha observado que técnicas obtienen los mejores resultados y como realizan el proceso de identificación. Finalmente se ha extraído en conjunto los mejores procesos identificados para la detección de manipulaciones y se han añadido variaciones para intentar mejorar los resultados obtenidos por estas investigaciones.

Los tres algoritmos propuestos en el presente trabajo han sido:

- Algoritmo basado en DCT junto a histogramas LBP para la detección de

manipulaciones.

- Algoritmo basado en DWT junto a histogramas LBP aplicando QMF para la detección de manipulaciones.
- Algoritmo de identificación de la región exacta duplicada en técnicas copia-pega.

Tras realizar numerosas experimentos para evaluar los resultados obtenidos se a concluido que el algoritmo que presenta mayor precisión es el algoritmo basado en DWT junto a histogramas LBP aplicando QMF. Este algoritmo consiguió una precisión máxima del 99,43 % en el conjunto de imágenes usado *CASIA v2.0*. A su vez demostró ser eficiente en las pruebas realizadas tras comparar su tiempo de ejecución con los demás algoritmos.

Por otro lado el algoritmo basado en DCT junto a histogramas LBP presenta resultados bajos de precisión para el conjunto de imágenes *CASIA v1.0*, por el contrario la precisión aumenta considerablemente en los otros datasets utilizados. Este algoritmo ha presentado mayor tiempo de ejecución en las pruebas realizadas debido al procesamiento que realiza con las imágenes.

En último lugar el algoritmo de identificación de la región exacta duplicada en técnicas copia-pega ha demostrado ser robusto y eficiente. En los resultados obtenidos se puede observar como consigue determinar la zona duplicada con gran precisión. Se han realizado pruebas con numerosas imágenes de diferentes texturas, dimensiones y formatos y en general ha identificado las áreas manipuladas correctamente. Este algoritmo presenta un tiempo de ejecución superior a los demás ya que hace uso de numerosos cálculos para determinar la zona duplicada.

6.2. Trabajo Futuro

Como posibles trabajos futuros pueden señalarse los siguientes:

- Investigar acerca de otras técnicas de identificación de manipulaciones

capaces de ser eficientes ante imágenes de grandes resoluciones captadas por dispositivos móviles de última generación.

- Desarrollar algoritmos para la identificación de manipulaciones realizadas mediante la técnica de empalme.
- Extender los algoritmos de identificación de manipulaciones a vídeos y no solo a imágenes digitales.
- Optimizar los algoritmos que hacen uso de la superposición tras dividir una imagen en bloques para conseguir realizar entrenamientos con mayor número de imágenes.

Capítulo 7

Introduction

7.1. Motivation

In the last few years the use of mobile devices has grown considerably, getting to be part of the daily life of society current. Because of this increase in devices, data traffic has also increased creating an ever larger communication network among the users that facilitates to share data massively and quickly. Between the shared data are the digital images that, thanks to social networks and instant messaging applications, have turned into one of the main focuses of data traffic.

The continuous improvement of the cameras incorporated in mobile devices close to the evolution of the image edition tools, have made it easier to manipulate an image with excellent results. To deal with this massive traffic of manipulated images, the area of forensic analysis investigates new manipulation detection techniques to evaluate the integrity of an image. The manipulated images are not only found in the traffic generated by the network, but have been around for decades and are present in many sectors (politics, cinema, press, etc.).

Due to the amount of information that a digital image can contain, it has been used for many years as the main method to influence society. That is why in the press and politics abounds a large amount of manipulated images. An example of this type of images with political purposes is shows in Figure **??** The prime Minister

of Canada, William Lyon, eliminated King Jorge VI from the original photograph. The manipulated image was used in a poster in the elections of the prime minister. On having eliminated the king of the image, it was giving a sensation of nearness to the village and not so much to the royal family.

(a) *Manipulated image* (b) *Original image*

Figura 7.1: Image manipulated for political purposes [FT17]

Not all manipulated images pursue political or ideological purposes, this type of images are very common in social networks where popularity is sought and reaches as many users as possible. An example of a viral photo that caused great repercussions, took place shortly after September 11, 2001. The image in Figure 7.2 was disseminated in the context of being the last photograph taken after the attack on the twin towers. Because of the attack day climate and several observations on the physical position from which the photograph was taken, it was shown that the picture has been manipulated.

In addition to the aforementioned sectors, the manipulated digital images have been important in other areas with very different objectives. For example, in the judicial sector, the images have been gaining importance because they can be an evidence of great value for the resolution of a trial. In order for an image could be used as valid proof or evidence of some act with legal purposes, its integrity must be ensured and demonstrated that it has not been manipulated. In order to carry

Figura 7.2: Viral manipulated image of the attack on the twin towers [Lis07]

out this type of authentication, it is necessary to make use of robust manipulation identification techniques that can guarantee, with great reliability, that the image is original.

As technology advances, it is increasingly difficult to detect this type of manipulation; tools like *Photoshop* or *GIMP* allow the edition of images with highly professional results. This type of software requires previous knowledge on the edition of images to obtain manipulations difficult to detect. With the increased use of mobile devices, data traffic and high-resolution embedded cameras, the applications that bring integrated image editing features are increasingly more. These applications, contrary to those mentioned previously, do not require any knowledge about image editing for getting manipulations very difficult to detect.

For all these reasons, the forensic analysis of digital images for mobile devices is very important these days. It is necessary to study and propose identification techniques that allow facing the large number of manipulated images that exist today.

7.2. Objectives

The objectives that have been marked in the present work are the following ones:

- Review the current literature regarding the uses of counterfeit images to obtain a detailed knowledge of the technologies of manipulation used today and their implementation.
- Study the state of the art on the techniques of detection of image manipulations.
- Design and implement robust and efficient algorithms that achieve optimal results in the detection of manipulations of digital images.
- Perform tests that demonstrate the robustness and validity of the results obtained by the algorithms.

7.3. Workplan

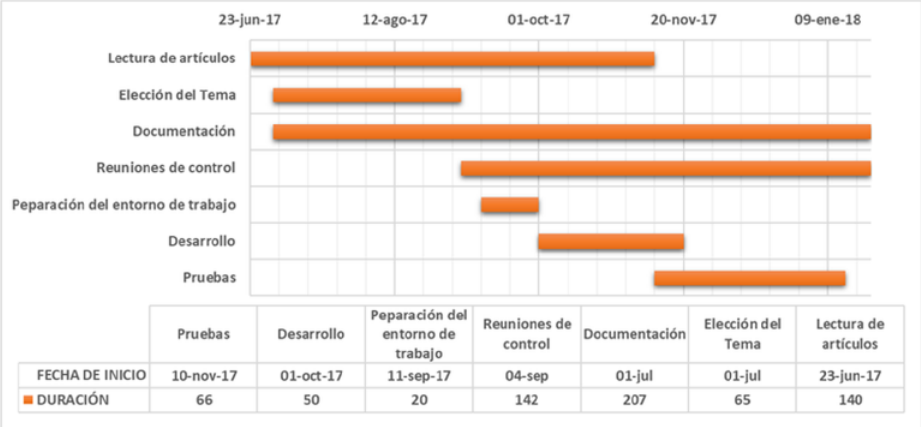
The work was divided into 7 phases. Then, a detailed explanation of each of the phases that have been carried out along the project.

- **Reading of articles**: This phase consisted in an investigation about: forensic image analysis, image falsification techniques and manipulation detection techniques. It was also reviewed the previous end of degree projects with the same theme common to ours, in order to obtain several views from other points of view.
- **Documentation**: The documentation activity was carried out in parallel throughout the whole project. In this way, possible losses of relevant information were avoided for the memory or to the development work.
- **Control meetings**: 2 sessions were arranged weekly to carry out a follow-up of the project. These are the following points to be made and shown the advances.
- **Preparation of the work environment**: This activity consisted of preparing the equipment with which we were going to work. We downloaded the appropriate tools as well as several checks in order that there were no

problems in the development phases. This activity was realized in major measure close to the meetings controls.

- **Development**: In this phase, we proceeded to carry out the development of the algorithms of this work. At the end of this activity the code tests were started, in this way the algorithms were improved until the most optimal results were obtained.
- **Tests**: This phase was the last to be carried out. There have been numerous quality tests of the results and efficiency of the algorithms. As can be seen in the Gantt diagram (7.3) this activity has taken more time than the development process. This is because it has been of great importance to contrast results with previous research.

To have a view of the work plan structured by dates, we can see the Gantt chart shown in Figure 7.3.



	Pruebas	Desarrollo	Peparación del entorno de trabajo	Reuniones de control	Documentación	Elección del Tema	Lectura de artículos
FECHA DE INICIO	10-nov-17	01-oct-17	11-sep-17	04-sep	01-jul	01-jul	23-jun-17
DURACIÓN	66	50	20	142	207	65	140

Figura 7.3: Gantt chart on the work plan

7.4. Structure of the Work

The work consists in 6 chapters in total, the current one being the introductory chapter: motivation, objectives, plan and its structure are described. Then, a brief explanation will be made of the other 5 chapters.

Chapter 2 takes a journey from the origins of photography to the present; examples of the evolution of manipulation techniques will be shown. Finally, the types of techniques and tools used in the manipulation of the digital images are detailed.

Chapter 3 describes the main digital image detection techniques manipulated, emphasizing the most relevant existing passive approach techniques in the literature. Reference will be made to investigations related to each of the existing manipulation techniques.

Chapter 4 deals with the contributions of this work. This chapter begins with some general considerations that will help to understand some specific details of the development part. The algorithms proposed in this work will be explained.

Chapter 5 presents the experiments that evaluate the algorithms proposed in the Chapter 4. It starts by detailing the configuration with which the tests have been performed and, later, it describes each of the experiments carried out: show the results and analysis of improvements and defects.

Finally, chapter 6 collects the final conclusions of the work. In the propose improvements for future work.

Capítulo 8

Conclusions and Future Work

8.1. Conclusions

Digital images contain a lot of relevant information. Due to this, they are a very important element in the legal field and they are evidences that provide great value in the resolution of a trial. In order that these evidences manage to be valid, it is necessary to be able to guarantee their authenticity and integrity in a reliable way. These days, there are numerous applications that manage to edit images with highly professional results. Detecting if an image has been modified by some manipulation technique is a complicated task. In order to guarantee the integrity of an image, it is very interesting to have forensic tools that can detect these falsifications.

In this work, an exhaustive study has been made on the existing techniques of manipulation detection, emphasizing in splice and copy - paste detection techniques.

After comparing the different investigations, the techniques that give the best results have been studied in depth, analyzing the process performed for the detection.

Finally, they have been designed and implemented in Python together with the best identified processes, a technique for detecting manipulations that improve the results obtained by these investigations.

The designed technique is composed of the following algorithms:

- Algorithm based in DCT with LBP histograms for manipulation detection.
- Algorithm based in DWT with LBP histograms applying QMF for manipulation detection.
- Algorithm of identification of the duplicated region in copy-paste techniques.

To evaluate the technique designed in this work, a set of experiments have been carried out with the same datasets used in the state of the art in order to compare the results. The results of the experiments are the following ones:

First, the algorithm that presents the most accuracy is the algorithm based on DWT together with LBP histograms applying QMF. This algorithm achieved a maximum accuracy of 99.43 % in the set of images used *CASIA v2.0*. In turn, it proved to be efficient in the tests carried out after comparing its execution time with the other algorithms. This is because it does not divide the image into blocks and works directly with the original image.

On the other hand, the algorithm based on DCT together with LBP histograms presents low precision results for the *CASIA v1.0* dataset. However, its accuracy increases considerably in the other datasets used. This algorithm has presented more execution time in the tests performed due to the processing that it carries out with the images.

The experiments with the detection algorithm of the region duplicated with copy-paste techniques have demonstrated their robustness and efficiency. In the results obtained it is possible to observe how the duplicated zone is located with great precision.

8.2. Future Work

As a possible future works, the following ones can be pointed out:

- Investigate other manipulation detection techniques capable of being efficient before images of high resolution caught by mobile devices of last generation.
- Develop algorithms for the identification of manipulations made by the splicing technique.

- Extend manipulation identification algorithms to videos and not just to digital images.furito5816 Optimize the algorithms that make use of the superposition after dividing an image into blocks to achieve training with a more number of images.

Bibliografía

[ABC+11] I. Amerini, L. Ballan, R. Caldelli, A. Del Bimbo, and G. Serra. A SIFT-Based Forensic Method for Copy-Move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, September 2011.

[AH17] A. Alahmadi and M. Hussain. Passive Detection of Image Forgery Using DCT and Local Binary Pattern. *Signal, Image and Video Processing*, 11(1):81–88, January 2017.

[AHA+13] A. A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, and G. Bebis. Splicing Image Forgery Detection Based on DCT and Local Binary Pattern. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 253–256, Riyadh, Saudi Arabia, December 2013.

[BJGY10] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei. Image Copy-Move Forgery Detection Based on SURF. In *2010 International Conference on Multimedia Information Networking and Security*, pages 889–892, Nanjing, China, November 2010.

[BSVB16] A. Bharati, R. Singh, M. Vatsa, and K. W. Bowyer. Detecting Facial Retouching Using Supervised Deep Learning. *IEEE Transactions on Information Forensics and Security*, 11(9):1903–1913, September 2016.

[CDR13] C. Chen, A. Dantcheva, and A. Ross. Automatic Facial Makeup Detection with Application in Face Recognition. In *2013 International Conference on Biometrics (ICB)*, pages 1–8, Madrid, Spain, June 2013.

[CIS17] CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, February 2017.

[CL11] C. C. Chang and C. J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[CPF04] A. C Popescu and H. Farid. January 2004.

[DW] J. Dong and W. Wang. CASIA TIDE v1.0 - v2.0. http://forensics.idealtest.org/.

[FSL03] J. Fridrich, D. Soukal, and J. Lukas. Detection of Copy Move Forgery in Digital Images. In *Proceedings of the Digital Forensic Research Workshop*, pages 5–8, Binghamton, New York, August 2003.

[FT17] Inc Fourandsix Technologies. Photo Tampering Throughout History. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, December 2017.

[FZWJ16] Q. Fu, X. Zhou, C. Wang, and B. Jiang. Mathematical relation between APBT-based and DCT-based JPEG image compression schemes. *Journal of Communications*, 11:84–92, January 2016.

[GVSORCHC17] L. J. García Villalba, A. L. Sandoval Orozco, J. Rosales Corripio, and J. Hernández Castro. A PRNU-based Counter-forensic Method to Manipulate Smartphone Image Source Identification Techniques. *Future Generation Computer Systems*, 76:418–427, November 2017.

[HGZ08] H. Huang, W. Guo, and Y. Zhang. Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, volume 2, pages 272–276, December 2008.

[IT14] IEEE IFS-TC. IFS-TC Image Forensics Challenge. http://ifc.recod.ic.unicamp.br/, January 2014.

[KAD15] N. Kose, L. Apvrille, and J. L. Dugelay. Facial Makeup Detection Technique Based on Texture and Shape Analysis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–7, Ljubljana, Slovenia, May 2015.

[KF11] E. Kee and H. Farid. A Perceptual Metric for Photo Retouching. In *Proceedings of the National Academy of Sciences, 108(50)*, pages 19907–19912, Hanover, USA, November 2011.

[KMC+07] N. Khanna, A. K. Mikkilineni, G. Chiu, J. P. Allebach, and E. Delp. Forensic Classification of Imaging Sensor Types. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6505, February 2007.

[KW08] X. Kang and S. Wei. Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics. In *2008 International Conference on Computer Science and Software Engineering*, volume 3, pages 926–930, December 2008.

[LFG06] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.

[Lis07] Listverse. Top 15 Photoshopped Photos That Fooled Us All. http://listverse.com/2007/10/19/top-15-manipulated-photographs/, October 2007.

[Mun17] El Mundo. Detenido por Circular a 200 Kilómetros por Hora tras Subir un Vídeo a Redes Sociales. http://www.elmundo.es/madrid/2017/08/30/59a68f0a468aeb7a658b4607.html, August 2017.

[RCAGSO+13] J. Rosales Corripio, D. M. Arenas González, A. L. Sandoval Orozco, L. J. García Villalba, J. C. Hernandez-Castro, and S. J. Gibson. Source Smartphone Identification Using Sensor Pattern Noise and Wavelet Transform. pages 1–6, Madrid, Spain, January 2013.

[SCC07] Y. Q. Shi, C. Chen, and W. Chen. A Natural Image Model Approach to Splicing Detection. In *Proceedings of the 9th workshop on Multimedia security*, pages 51–62, Dallas, Texas, September 2007.

[SOAGRC+14] A. L. Sandoval Orozco, D. M. Arenas González, J. Rosales Corripio, L. J. García Villalba, and J. C. Hernandez-Castro. Source Identification for Mobile Devices, Based on Wavelet Transforms Combined with Sensor Imperfections. *Computing*, 96(9):829–841, September 2014.

[Sun14] Vancouver Sun. Photos: 20 More Stars and Celebrities Before and After Photoshop. http://www.vancouversun.com/life/fashion-beauty/Photos+more+stars+celebrities+before+after+Photoshop/7841314/story.html, July 2014.

[WDT10] W. Wang, J. Dong, and T. Tan. Image Tampering Detection Based on Stationary Distribution of Markov Chain. In *2010 IEEE International Conference on Image Processing*, pages 2101–2104, Hong Kong, China, September 2010.

[XYS+16] Z. Xia, C. Yuan, X. Sun, D. Sun, and R. Lv. Combining wavelet transform and LBP related features for fingerprint liveness detection. *IAENG International Journal of Computer Science*, 43(3):290—-298, April 2016.

[ZKR08] Z. Zhang, J. Kang, and Y. Ren. An Effective Algorithm of Image Splicing Detection. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 1035–1039, December 2008.

[ZL11] X. Zhao and J. Li. Detecting Digital Image Splicing in Chroma Spaces. In *Digital Watermarking*, volume 6526, pages 12–22, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[ZWWZ17] Z. Zhang, D. Wang, .C Wang, and X. Zhou. Detecting Copy-move Forgeries in Images Based on DCT and Main Transfer Vectors. *KSII Transactions on Internet and Information Systems*, 11:4567–4587, September 2017.

[ZZ12] Y. Zhang and C. Zhao. Revealing Image Splicing Forgery Using Local Binary Patterns of DCT Coefficients. In *Communications, Signal Processing, and Systems*, January 2012.